

2
2009

LIBRARY
Michigan State
University

This is to certify that the
dissertation entitled

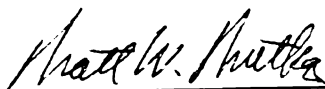
Rate Allocation and QoS Support in Wireless Mesh Networks

presented by

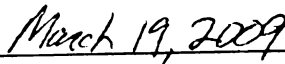
Bo Wang

has been accepted towards fulfillment
of the requirements for the

Ph.D. degree in Computer Science



Major Professor's Signature



Date

PLACE IN RETURN BOX to remove this checkout from your record.
TO AVOID FINES return on or before date due.
MAY BE RECALLED with earlier due date if requested.

DATE DUE	DATE DUE	DATE DUE

RATE ALLOCATION AND QOS SUPPORT IN WIRELESS MESH NETWORKS

By

Bo Wang

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Computer Science

2009

ABSTRACT

RATE ALLOCATION AND QOS SUPPORT IN WIRELESS MESH NETWORKS

By

Bo Wang

Wireless mesh networking (WMN) is an emerging technology that uses multi-hop communications to provide cost-efficient broadband Internet access for community or enterprise users. With the increasing popularity of content distribution and multimedia applications, efficient rate allocation and QoS support become essential for the wide deployment of wireless mesh networks.

In this dissertation we discuss several research topics related to rate allocation and QoS support in wireless mesh networks. First we propose a framework named QUOTA for QoS support and fair rate allocation in CSMA/CA based wireless mesh networks. Our framework uses contention graphs and Network Utility Maximization (NUM) to perform admission control and rate allocation. Simulation results show that our framework successfully guarantees the QoS of real-time flows and fairly, efficiently allocates bandwidth for elastic flows in different wireless mesh network scenarios.

Second we propose a framework that performs rate allocation for multicast and unicast traffic in TDMA based wireless mesh networks using Network Utility Maximization. In addition, a graph coloring based scheduling algorithm is proposed to realize the allocated rates. Simulation results show that our framework provides guar-

anteed throughput and low delay for both multicast and unicast traffic. Furthermore, our framework significantly outperforms a previously published framework that has a similar objective.

Third we propose a routing algorithm for minimum length schedule in multi-channel TDMA based wireless mesh networks. We consider network scenarios where multiple orthogonal channels are available. With a channel assignment algorithm to eliminate secondary interference, we are able to use a scheduling algorithm that yields the minimum length schedule given a specific routing tree. We then propose a heuristic routing algorithm that aims to build the routing tree that results in the minimum length schedule. Our routing algorithm performs significantly better than simple routing algorithms, which are based on Breadth First Search or Dijkstra algorithms.

Fourth we propose a path selection metric for mobile stations in IEEE 802.16 Mobile Multihop Relay (MMR) networks to select appropriate paths to the base station. The proposed path selection metric, Normalized Number of Minislots (NNM), effectively takes all the factors that affect the end-to-end achievable throughput into consideration. Simulation results demonstrate that our metric yields better performance in terms of throughput and delay compared to existing path selection metrics, especially when networks have high traffic load.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Professor Matt Mutka for his patient supervision. Without Professor Mutka's scientific guidance and financial support, this dissertation could not have come into light.

I gratefully acknowledge my dissertation committee, Professor Philip McKinley, Professor Pang-Ning Tan, and Professor Hayder Radha for their valuable guidance and comments. Special thanks go to Dr. Jonathan Shapiro, Professor Eric Torng and Professor Li Xiao for their valuable advisory.

Many thanks go to my colleagues in eLANS lab and CSE department. Special thanks go to Guokai Zeng and Sohraab Soltani for their help on various research topics.

I am grateful to my parents and my wife Fei Lu for supporting me all the time.

TABLE OF CONTENTS

LIST OF TABLES	viii
LIST OF FIGURES	ix
1 Introduction	1
1.1 Background	1
1.2 Motivation	5
1.3 Structure of the Content	7
2 Related Work	8
2.1 Research Challenges in Wireless Mesh Networks	8
2.1.1 Capacity	9
2.1.2 Routing	11
2.1.3 Management	13
2.2 Rate Allocation	15
2.3 QoS Support	16
2.3.1 QoS Support in CSMA/CA Based Wireless Mesh Networks . .	16
2.3.2 TDMA Scheduling	17
2.3.3 Routing Algorithms for Minimum Length Schedule	19
2.3.4 Path Selection in MMR Networks	20
3 QoS-Aware Fair Rate Allocation in CSMA/CA Based Wireless Mesh Networks	23
3.1 Theoretical Framework	24
3.1.1 Link Contention Graph and Rate Allocation Feasibility	26
3.1.2 Admission Control	30
3.1.3 Utility Maximization Based Rate Allocation	31
3.2 Framework Implementation and Practical Issues	35
3.2.1 Distributed Clique Construction	36
3.2.2 Centralized Admission Control and Rate Allocation	40
3.2.3 Adaptive Effective Clique Capacity	41
3.3 Performance Evaluation	42
3.3.1 Simulation Setup	43
3.3.2 Simulation Results of a Single-branch Scenario	44
3.3.3 Simulation Results of Multi-branch Scenarios	49
3.4 Summary	54

4	Optimization Based Rate Allocation and Scheduling in TDMA Based Wireless Mesh Networks	55
4.1	System Model	57
4.1.1	Routing Tree and Transmission Set	57
4.1.2	Transmission Contention Graph	60
4.1.3	Clique Feasibility and Scheduling Feasibility	62
4.2	Rate Allocation and Scheduling Framework	67
4.2.1	Least Overlapped First	67
4.2.2	Utility Maximization Based Rate Allocation	69
4.2.3	Optimal Graph Coloring Based Scheduling	71
4.3	Framework Extensions	74
4.3.1	Real-time Session Rate Allocation	74
4.3.2	Channel Assignment	75
4.4	Performance Evaluation	76
4.4.1	Real-time Traffic Rate Allocation Evaluation	77
4.4.2	Simulation Results for Single-channel Scenarios	79
4.4.3	Simulation Results for Multi-channel Scenarios	82
4.5	Summary	84
5	Routing for Minimum Length Schedule in Multi-channel TDMA Based Wireless Mesh Networks	89
5.1	Preliminaries	90
5.1.1	System Model	90
5.1.2	Joint Routing and Scheduling Optimization	92
5.1.3	Optimal Link Scheduling	94
5.2	Routing for Minimum Length Schedules	95
5.2.1	Heuristic Routing Algorithms	96
5.2.2	Algorithm Illustration	100
5.3	Further Discussions	101
5.3.1	Channel Assignment	101
5.3.2	Multiple Gateways	103
5.4	Performance Evaluation	104
5.4.1	Simulation Setup	104
5.4.2	Simulation Results	105
5.5	Summary	108
6	Path Selection for Mobile Stations in IEEE 802.16 Multihop Relay Networks	110
6.1	Preliminaries	111
6.1.1	Frame Structure	111
6.1.2	Fast Access Station Switching	112

6.2	Path Selection Metric	113
6.2.1	Number of Hops (NOH)	114
6.2.2	Maximum End-to-End Throughput (MET)	114
6.2.3	Normalized Number of Minislots (NNM)	115
6.3	Performance Evaluation	119
6.3.1	Simulation Setup	119
6.3.2	Simulation Results	121
6.4	Summary	125
7	Conclusions and Future Work	128
7.1	Conclusions	128
7.2	Future Work	130
7.2.1	Real System Evaluation	130
7.2.2	User-deployed Wireless Mesh Network	130
7.2.3	Heterogeneous Wireless Networks	131
	BIBLIOGRAPHY	132

LIST OF TABLES

3.1	Performance Comparison of the Single-branch Scenario (CBR Real-time Traffic)	48
3.2	Performance Comparison of the Single-branch Scenario (VBR Real-time Traffic)	49
4.1	An Example Transmission Set	60
4.2	Simulation Parameters	78
4.3	Modulation Type and Link Rate vs. SNR	78
4.4	Session List	79
5.1	Simulation Parameters	106
5.2	Link Rate vs. SNR	106
6.1	Simulation Parameters	120

LIST OF FIGURES

1.1	Wireless Mesh Network Scenario	2
1.2	IEEE 802.16 Mobile Multihop Relay Network	4
2.1	A Downlink Routing Tree Example in WiMAX Mesh Networks . . .	18
2.2	A Downlink Routing Tree Example in MMR Networks	21
3.1	A Branch of the Mesh Network	25
3.2	Link Contention Graph of the Mesh Network Scenario Shown in Figure 3.1	26
3.3	Local Link Contention Graph Construction Example	38
3.4	Distributed Clique Construction Example. The Associated Network Topology is Shown in Figure 3.1; The Associated Global Link Contention Graph is Shown in Figure 3.2.	39
3.5	A Single-Branch Scenario	44
3.6	Allocated Rate of Different Flows at Different Rounds under QUOTA	45
3.7	Throughput of Different Flows at Different Rounds under QUOTA . .	46
3.8	Average Delay of Two Real-time Flows at Different Rounds under QUOTA	47
3.9	Aggregate Throughput of Network Flows	50
3.10	Average Delay of Real-time Flows	51
3.11	Average Fairness Index of Branches	52
3.12	QUOTA Control Message Overhead	53

4.1	An Example Scenario	58
4.2	An Example Contention Graph with both Primary and Secondary Interference	61
4.3	An Example Contention Graph with only Primary Interference	61
4.4	An Example Link Contention Graph	64
4.5	The Schedule Obtained by LOF	69
4.6	The Schedule Obtained by OGC	74
4.7	MAC Frame Structure	77
4.8	Simulation Scenario	79
4.9	Throughput vs. Traffic Request	80
4.10	Aggregated Throughput vs. Number of Sessions	81
4.11	Average Delay vs. Number of Sessions	82
4.12	Total Utility vs. Number of Sessions	83
4.13	Aggregated Throughput vs. Number of Time Slots	84
4.14	Average Delay vs. Number of Time Slots	85
4.15	Total Utility vs. Number of Time Slots	85
4.16	Aggregated Throughput vs. Number of Sessions (Multiple Channels)	86
4.17	Average Delay vs. Number of Sessions (Multiple Channels)	86
4.18	Total Utility vs. Number of Sessions (Multiple Channels)	87
4.19	Aggregated Throughput vs. Number of Time Slots (Multiple Channels)	87

4.20	Average Delay vs. Number of Time Slots (Multiple Channels)	88
4.21	Total Utility vs. Number of Time Slots (Multiple Channels)	88
5.1	An Example Scenario	96
5.2	Routing Trees	97
5.3	An Illustration Scenario	101
5.4	Modified Dijkstra Algorithm Illustration	102
5.5	Routing Algorithm Illustration	103
5.6	Number of Time Slots vs. Number of Sessions (Gateway at the Center)	107
5.7	Number of Time Slots vs. Number of Mesh Routers (Gateway at the Center)	108
5.8	Number of Time Slots vs. Number of Sessions (Gateway at the Corner)	109
5.9	Number of Time Slots vs. Number of Mesh Routers (Gateway at the Corner)	109
6.1	IEEE 802.16j MAC Frame Structure	112
6.2	Simulation Topology	119
6.3	Throughput vs. Traffic Rate	121
6.4	Delay vs. Traffic Rate	122
6.5	Downlink Satisfaction Index vs. Traffic Rate	123
6.6	Downlink Load Ratio vs. Traffic Rate	124
6.7	Throughput vs. Number of Mobile Stations	125

6.8	Delay vs. Number of Mobile Stations	126
6.9	Downlink Satisfaction Index vs. Number of Mobile Stations	126
6.10	Downlink Load Ratio vs. Number of Mobile Stations	127

CHAPTER 1

Introduction

1.1 Background

Wireless mesh networking (WMN) is a promising technology for building broadband wireless access networks. Figure 1.1 shows a wireless mesh network. In this typical scenario, the network contains mesh routers and mesh clients [1]. Mesh routers have limited mobility and they are connected to form a multi-hop backbone for the network. Some of the mesh routers are connected to the Internet, which become Internet gateways. Mesh clients, such as laptops and PDAs, connect to mesh routers to access the Internet and share network resources.

Mesh networks have been proposed for several application areas. Some examples of mesh network applications are as follows [1]:

- *Home networking.* A popular home networking technology is Wireless Local Area Networks (WLAN). An Access Point (AP) is deployed to provide Internet

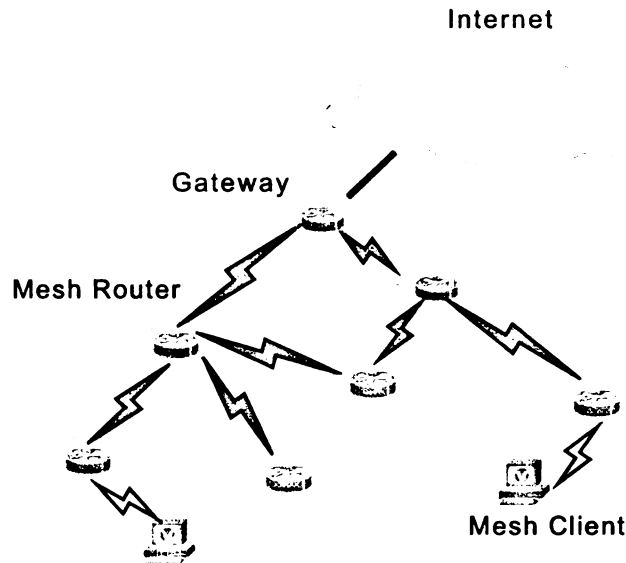


Figure 1.1. Wireless Mesh Network Scenario

access for various wireless devices in a house. However, a single access point may not provide full coverage for the house and multiple access points need to be connected through a switch which increases the deployment cost. Wireless mesh networking enables access points to be connected through wireless links. Only a single connection to the Internet is needed. Wireless mesh networking can therefore improve network coverage and reduce deployment cost compared with WLAN technology.

- *Community and neighborhood networking.* Wireless mesh networks provide a cost-efficient way for the communities to access Internet and share network resources. Recently several community networks based on wireless mesh technologies have emerged [2, 3, 4]. The typical deployment of community mesh networks would be to mount wireless routers outside of windows or on roofs [5]. Client devices are connected to mesh routers. Some mesh routers are connected

to the Internet so the whole community can share the Internet access. In addition, clients in the community mesh networks can share network resources or interact with each other without going through the Internet.

- *Security surveillance systems.* There are several advantages of using wireless mesh technology in security surveillance systems. First, wireless mesh technology provides a more practical way than wired networks to connect several surveillance devices with low cost. Second, dedicated mesh backbone facilitates the transmission of surveillance information which needs high bandwidth and high reliability.

For the ease of presentation, we classify wireless mesh networks according to the underlying physical layer properties and link layer protocols. IEEE 802.11 [6] defines a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) MAC protocol. We denote mesh networks based on the IEEE 802.11a/b/g standards as CSMA/CA based wireless mesh networks.

IEEE 802.16 [7] (commonly known as WiMAX) employs Time Division Multiple Access (TDMA) MAC protocol to support collision-free communications. IEEE 802.16 supports two modes of operations: Point-to-Multipoint (PMP) mode and mesh mode. IEEE 802.16-2004 standard defines physical layer properties and MAC protocols for both PMP and the mesh mode. However, it does not support mobility. We denote mesh networks based on IEEE 802.16-2004 standard as WiMAX mesh networks.

Recently, IEEE 802.16j [8] has been proposed to support mobility in mesh mode.

IEEE 802.16j based wireless mesh networks are usually named Mobile Multihop Relay (MMR) networks. Figure 1.2 shows a typical IEEE 802.16 MMR network consisting of base stations (MR-BS), relay stations (RS) and mobile stations (MS). We denote MMR cell as the radio coverage area of a base station and all its subordinate relay stations where mobile stations are serviced [9]. In Figure 1.2, only the coverage areas of base stations are shown. The base station is the gateway for data traffic of mobile stations from or to the Internet. Communication resources within a cell are managed by the base station through routing and scheduling. The relay stations are used to extend the coverage of the cell and to enhance the throughput of mobile stations.

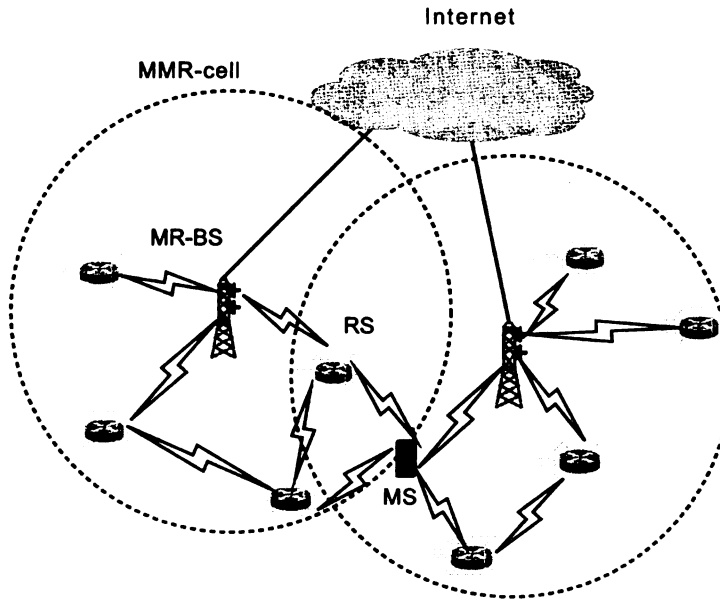


Figure 1.2. IEEE 802.16 Mobile Multihop Relay Network

1.2 Motivation

Flows within wireless mesh networks may be classified as real-time and elastic [10]. Real-time flows include video and voice applications that have requirements on the transmission rate and are very sensitive to latency and jitter. For example, voice over the mesh requires that the one-way latency to be smaller than 200 milliseconds (ms) and jitter should be less than 50 ms [11]. Elastic flows are from applications that can adjust their transmission rate gradually. Such applications include file transfer, email and remote terminal [10].

Two major challenges for the wide deployment of wireless mesh networks are fair rate allocation for elastic flows and QoS support for real-time flows. It is well known that using TCP and the CSMA/CA MAC protocol (e.g., IEEE 802.11 [6]) in multi-hop wireless networks results in severe unfairness [12]. In addition, the randomness nature of CSMA/CA MAC protocols makes QoS support inherently difficult. Spatial TDMA has become a popular alternative MAC technology for wireless mesh networks. By supporting collision-free communications, TDMA based MAC protocols enable more accurate resource allocation and more flexible QoS support. In TDMA based wireless mesh networks, routing and scheduling algorithms are essential for satisfying the rate allocation and QoS requirements. In this dissertation, we propose a set of algorithms to perform efficient, fair rate allocation for elastic flows and provide QoS support for real-time flows in both CSMA/CA and TDMA based wireless mesh networks.

In previous work, Quality-of-Service (QoS) support and fair rate allocation are usually considered separately. Fair rate allocation frameworks proposed in previous

work focus on elastic flows that do not have explicit service quality requirements. Such frameworks may not be applied directly within mesh networks containing real-time flows that have explicit rate and delay requirements. QoS support frameworks for wireless networks guarantee service quality of real-time flows by reserving bandwidth for them. However, the major drawbacks of existing approaches are the inefficiency and the lack of fairness consideration in bandwidth allocation, which results in poor performance of elastic flows. In this dissertation, we propose QUOTA [13], a framework that combines QoS support and fair rate allocation in CSMA/CA based wireless mesh networks.

With the increased popularity of multicast and multimedia applications, efficient rate allocation and scheduling algorithms that provide guaranteed throughput and bounded delay for both multicast and unicast traffic become very important. In this dissertation, we propose a framework that performs rate allocation for multicast and unicast traffic in TDMA based wireless mesh networks using Network Utility Maximization [14]. In addition, a graph coloring based scheduling algorithm is proposed to realize the allocated rates.

In TDMA based wireless mesh networks, routing algorithms have great impact on the performance of the scheduling algorithms and are essential to provide QoS support for mesh clients. In order to maximize the network throughput and minimize the session delay, the routing and scheduling algorithms should produce a minimum length schedule. We solve the above problem for network scenarios where multiple orthogonal channels are available. With a channel assignment algorithm to eliminate secondary interference, we are able to use a scheduling algorithm that yields the

minimum length schedule given a specific routing tree. We then propose a heuristic routing algorithm that aims to build the routing tree that results in the minimum length schedule.

In IEEE 802.16 MMR networks, choosing an appropriate path between any mobile station and the base station is the basis for providing QoS support for real-time applications. In this dissertation, we propose a path selection metric, named Normalized Number of Minislots (NNM) [15], which enables a mobile station to choose a path that satisfies its application rate and delay requirements.

1.3 Structure of the Content

The remainder of this dissertation is organized as follows: In Chapter 2 we review the related work. Chapter 3 introduces the QUOTA framework for CSMA/CA based wireless mesh networks. We discuss the rate allocation and scheduling framework for TDMA based wireless mesh networks in Chapter 4. Chapter 5 proposes a routing algorithm that aims to produce minimum length schedule in multi-channel TDMA based wireless mesh networks. In Chapter 6 we present a path selection metric for mobile stations in IEEE 802.16 MMR networks. Finally, a summary and possible future work are given in Chapter 7.

CHAPTER 2

Related Work

In this chapter, we first give a general overview on the research challenges and proposed solutions for wireless mesh networks. Then we present previous work related to rate allocation and QoS support in wireless mesh networks.

2.1 Research Challenges in Wireless Mesh Networks

Wireless mesh networks share many features with wireless ad hoc networks such as being multi-hop, lack of wired infrastructure, etc [1]. However, wireless mesh networks have some special properties and therefore present research challenges that are not present in ad hoc networks [1].

2.1.1 Capacity

The primary application of wireless mesh networks is providing high-bandwidth connectivity for community and enterprise users. However, due to inter-channel interference, the capacity of a multi-hop network quickly degrades as the node density and network diameter increases [16, 17, 18]. Thus, a major challenge for wireless mesh networks is achieving sufficient network capacity to meet the requirement of applications such as high-speed Internet access and video sharing among community members.

IEEE 802.11 standards [6] divide available radio frequency to several channels with adjacent channels overlapping in frequency. In wireless networks, when several transmissions in close proximity use the same or overlapping channels, inter-channel interference can corrupt data and lower the effective capacity of the network. Only non-overlapping (orthogonal) channels can eliminate inter-channel interference. IEEE 802.11b provides 3 non-overlapping channels in the 2.4 GHz spectral band and IEEE 802.11a provides 13 non-overlapping channels in the 5 GHz band [19].

Having multiple orthogonal channels creates the possibility of increasing network capacity through the promising approach of multi-channel multi-radio communication. In this approach, each wireless device in the network may have several, say m , wireless radios (interfaces) and communicates with other devices using a set of c wireless channels. If each node has a dedicated interface for every possible channel, i.e., $m = c$, the network capacity can achieve a c -fold increase since c interfaces can transmit simultaneously using orthogonal channels. However, the number of in-

terfaces usually is smaller than the number of channels ($m < c$) due to the cost of wireless interfaces. In this case, at least an m -fold increase can readily be achieved by fixing m interfaces on m different channels [16]. Moreover, Kyasanur and Vaidya show that, when c/m is $O(\log n)$, a c -fold increase can still be obtained in a random network with n nodes [20]. Considering the small number of available channels in typical IEEE 802.11 wireless networks, the above result has practical interest due to the possibility of achieving optimal network capacity with just few interfaces per node [20].

There are several ways to improve network capacity in addition to multi-channel communication. One possible way is using directional antennas [21, 22]. Omnidirectional antennas transmit roughly equally in all directions due to the uniform scattering of radio signals. A directional antenna, on the other hand, restricts the propagation of radio signals to one or two directions [21]. When omnidirectional antennas are used, all nodes in a roughly elliptical vicinity around a pair of directly communicating nodes must stay silent. However, when directional antennas are used, several pairs of nodes in close proximity can communicate simultaneously provided the intended recipients do not lie in overlapping transmission cones. The resulting simultaneous communications effectively increase the network capacity.

Another possible solution is to use a technique called spatial backoff to improve the utilization of any given channel [16]. Due to channel interference, every node in a network has a contending area where other nodes in this area can not transmit at the same time [16]. The smaller the contending area, the more concurrent transmission are enabled. Several techniques have been proposed to improve the network capacity

by adjusting the contending area dynamically using transmission power control and carrier sense threshold adjustment [23, 24].

2.1.2 Routing

Routing protocols in wireless mesh networks are similar to those in ad hoc networks in terms of protocol architecture. In fact, currently deployed wireless mesh networks often use routing protocols based on some of the most popular ad hoc routing protocols. For example, the Microsoft mesh testbed [25] uses a modified version of the Dynamic Source Routing (DSR) [26] protocol. The Kiyon network [27] uses Ad hoc On-demand Distance Vector (AODV) [28] as a basis for its routing protocols. However, routing protocols such as AODV and DSR were designed to support networks where nodes have high mobility. In such networks, quickly adapting to link changes is the primary task. Because links in wireless mesh networks tend to be long lived, routing protocols can be more concerned about finding reliable and high throughput paths. Consequently, ad hoc routing protocols must be modified to achieve higher levels of performance that are possible in mesh networks.

Traditional ad hoc routing protocols such as AODV and DSR use the hop count as routing metric. However, this metric is not suitable for a wireless mesh network for several reasons.

First, hop count metric fails to take link quality into consideration. The link quality can be defined in terms of measurable quantities such as loss rate, bandwidth, etc. The idea that minimizing hop count will also minimize packet delay and maximize

network throughput is based on the assumptions that a link either works perfectly or not at all and that all links have uniform bandwidth. In real wireless mesh networks, however, these assumptions are routinely violated. Link level measurements in wireless mesh networks reveal that links typically have an intermediate and asymmetric loss rate [29, 30]. It often happens that a two-hop route with high quality links exists and provides higher throughput than a single lossy link. However, routing protocols using the hop count metric will always prefer the one-hop route. Dravels, et al. [31] proposed Link Quality Source Routing (LQSR) as an extension to DSR. The primary difference between LQSR and DSR is the incorporation of a new routing metric named Expected Transmission Count (ETX). The ETX metric, originally proposed by Couto et al., is designed to favor routes with high end-to-end throughput [32]. The metric is defined as the expected number of transmissions (including retransmission) needed to successfully deliver a packet over a link.

The IEEE 802.11 standards adaptively select a physical layer bit-rate based on the observed channel quality [33]. This widely used technique called multi-rate radio makes the link bandwidth non-uniform in multi-hop networks. From the perspective of routing, the physical properties of communication channels presents a fundamental tradeoff when multi-rate radio is used. Because high link rates require high channel quality and because channel quality deteriorates with distance, high rates can only be sustained over short communication ranges, whereas low rate links can operate over long communication ranges [34]. Consequently, the selection of minimum hop paths often results in routes whose links operate at low transmission rates and with high loss rates [34]. As the goal of wireless mesh networks is to provide high bandwidth

network access, such routes are far from optimal. Awerbuch, et al. [34] proposed a link metric named Medium Time Metric (MTM) for multi-rate ad hoc networks. Each link is assigned a weight that is proportional to the amount of medium time used to transmit a packet with certain size on that link. The path metric is the sum of weights of all links on the path. Consequently, paths that minimize the total consumed medium time will be favored. The MTM metric reflects the link bandwidth since higher rate links consumes less medium time for sending packets. It is worth noting that because MTM does not account for link losses, it can be combined with a complementary metric such as ETX, which is solely dependent on loss rate [35, 32].

In addition, the hop count metric fails to take channel diversity into consideration. Channel diversity refers to the quality of a route with respect how many orthogonal channels are used. Draves, et al. [35] proposed Weighted Cumulative Expected Transmission Time (WCETT) routing metric for multi-radio multi-channel wireless mesh networks. The metric is designed to satisfy three requirements: First, the link quality should take both loss rate and bandwidth into consideration. Second, adding a link to a path should always increase its cost. Finally and most important, the metric should account for the effects of channel interference on throughput.

2.1.3 Management

Although ad hoc networks are generally difficult to manage due to mobility and resource constraints of wireless devices, the existence of a wireless mesh backbone provides several advantages that can be exploited to facilitate the management task.

First, mesh routers have limited mobility and abundant resources such as energy, storage and processing power and can therefore be used as dedicated devices to perform management tasks. In addition, a high throughput and reliable mesh backbone facilitates the aggregation of monitoring information.

For mesh networks deployed by the Internet Service Provider (ISP), troubleshooting is very important for ensuring the reliable and efficient operation of large-scale wireless mesh backbones [36]. However, it is also challenging given the complexity and unpredictability of several factors such as traffic pattern, radio propagation, etc. that affect the network performance [36]. A general troubleshooting framework consists of two components: data collection and data processing. In ISP-deployed mesh networks, all mesh routers can perform data collection task but processing is delegated to one or a few central servers located at certain mesh routers. Such a framework is referred to as centralized troubleshooting. Qiu, et al. [36] proposed a troubleshooting framework for wireless mesh networks. All nodes in the network play the role of Agent by continuously collecting and reporting data to a Manager server. The Manager server cleans the data and runs a root cause analysis to locate network anomalies and faults. The Manager server can be deployed at a single node or at a set of nodes.

For user-deployed mesh networks, mesh routers are maintained by different persons and organizations. Consequently, full cooperation in routing and data forwarding can not be assumed a priori. Simulation and testbed experiments show that non-cooperative behaviors greatly affect the network performance in terms of throughput and latency [37, 38]. Thus cooperation enforcement is of primary importance for user-

deployed mesh networks. Since it is unlikely that any user-controlled mesh routers can be dedicated to a management task, cooperation enforcement operations are ideally conducted in a distributed fashion. A seminal paper in the area of misbehavior detection is by Marti et al. [39]. The authors proposed Watchdog and Pathrater agents in the network. Buchegger et al. [40, 41] proposed the CONFIDANT system to deal with the data forwarding problem in wireless ad hoc networks. Mahajan et al. [38] proposed the Catch protocol. Catch is designed to solve the following two problems: differentiate selfish behavior such as dropping packets from the case where a node fails to receive the packet; make the neighbors of selfish nodes punish them by simultaneously isolating them. By doing this, selfish nodes are forced to cooperate by forwarding data. The major novelty of the Catch protocol is the use of anonymous messages. Although naive selfish strategies based on dropping data packets are readily detected, more sophisticated strategies that manipulate ad hoc routing protocols present a greater challenge. Wang, et al. [42, 43] proposed a method to distinguish selfish peers from cooperative ones based solely on local observations of AODV routing protocol behavior.

2.2 Rate Allocation

The Network Utility Maximization (NUM) framework has been used for fair rate allocation of elastic flows in several research papers. This framework was originally proposed for Internet congestion control [44, 45] and recently has been used for fair rate allocation in wireless networks [46, 47, 48, 49]. Some of them [47, 48] focus on

rate allocation in single hop wireless networks. Others [46, 50, 49] pay more attention to multi-hop wireless networks.

In the NUM framework, a utility function $U_s(x_s)$ is associated to every network session $s \in S$ in the network [10] where x_s is the rate of session s . The utility function can be considered as a reflection of user's satisfaction [10]. For elastic flows, it is usually assumed to be increasing, strictly concave and twice continuously differentiable [46, 47]. It has been shown that [44, 51], by maximizing $\sum_{s \in S} U_s(x_s)$, a certain targeted fairness is achieved among elastic flows. Furthermore, if the utility function is defined as $U_s(x_s) = \ln(x_s)$, proportional fairness is achieved [44, 51]. Depending on the modeling of the problem, the constraints of the maximization problem maybe be convex or non-convex. For convex constraints, the maximization problem can be solved efficiently and distributely with the help of Lagrange duality theory [52]. For non-convex constraints, the problem is difficult to solve, especially in a distributed way [53].

2.3 QoS Support

2.3.1 QoS Support in CSMA/CA Based Wireless Mesh Networks

QoS support in CSMA/CA based wireless mesh networks is usually achieved through admission control, rate policing and packet level scheduling.

SWAN [54] framework provides service guarantee for real-time flows by control-

ling the rate of elastic flows. The rate controller is deployed between the IP and the MAC layer. The flow rate control algorithm used in SWAN is additive increase, multiplicative decrease (AIMD). When the MAC layer detects packet delay that exceeds a threshold, which is determined by the delay requirement of the real-time flows, the rate control algorithm is triggered and a new transmission rate is determined. The rate adjustment is achieved through a leaky bucket traffic shaper.

QPART [55] is a distributed framework to provide QoS guarantees for real-time flows. QPART uses priority-based admission control and conflict-aware packet scheduling to ensure that the requirements of real-time flows can be accommodated. MPARC [56] provides throughput guarantee for multi-priority traffic through accurate admission control for real-time flows and appropriate rate policing on elastic flows. DDA [57] provides average delay guarantees to real-time flows in ad hoc networks by adapting IEEE 802.11 parameters such as the contention window size.

2.3.2 TDMA Scheduling

Spatial TDMA is a generalization of the TDMA protocol for multi-hop networks and has been proposed in [58]. Several papers have adopted the spatial TDMA protocol for scheduling, both for unicast [59] and multicast traffic [60, 61].

In this dissertation, we focus on WiMAX mesh networks with TDMA scheduling. In WiMAX mesh networks, the base station (BS) acts as the Internet gateway and is responsible for constructing the downlink and uplink routing tree [62]. The mesh routers are also called subscribe stations (SS). Figure 2.1 shows an example downlink

routing tree. Lines in the figure represent the links for downlink traffic. Note that the uplink routing tree may be different than the downlink routing tree.

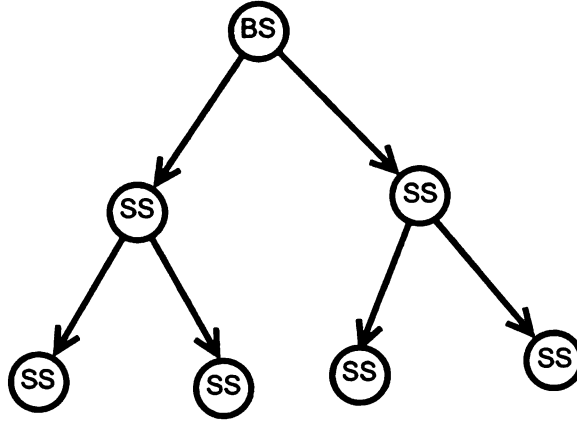


Figure 2.1. A Downlink Routing Tree Example in WiMAX Mesh Networks

The base station needs to translate the traffic rate requirements of subscribe stations into the number of time slots that are needed for each link in a frame (scheduling period). The number and the order of the time slots assigned to each link is determined by the scheduling algorithm. Several scheduling algorithms have been proposed [62, 63, 64, 65, 66, 67] for WiMAX mesh networks.

IEEE 802.16 [7] uses an example to propose a simple scheduling algorithm. The algorithm assigns time slots to links following a breadth-first traversal of the routing tree. Links that are visited first are assigned time slots earlier in the frame. When the total number of time slots assigned to links exceeds the number of available time slots in a frame, the algorithm scales down the number of time slots assigned to each link. The scheduling algorithm is performed at the beginning of each frame period at the base station.

An interference-aware scheduling algorithm is proposed in [64]. The proposed

algorithm aims to find the maximum number of concurrent transmissions to achieve high throughput. The algorithm assigns time slots in rounds. In each round, the algorithm assigns one time slot to the link with the highest unallocated traffic demand, and all other links that do not interfere with it. The allocation terminates when no times slots left in a frame or all links are scheduled.

A load-balancing algorithm is proposed in [65] and is improved in [62]. At the beginning of each frame, the algorithm calculates the satisfaction index of each link, which is defined as the ratio of the assigned rate over the requested rate in the last frame. Links with lowest satisfaction index are scheduled first. Links that get truncated at the end of one frame will be scheduled earlier in the next frame.

Djukic, et al. propose the Bellman-Ford TDMA scheduling algorithm in [63, 67]. The major advantage of Bellman-Ford scheduling algorithm over the interference-aware scheduling algorithm and the load-balancing algorithm is that it takes the scheduling delay into consideration while taking advantages of spatial reuse.

2.3.3 Routing Algorithms for Minimum Length Schedule

In order to maximize the network throughput and minimize the session delay in TDMA based wireless mesh networks, the routing and scheduling algorithms should produce a minimum length schedule. A linear programming formulation of the joint routing and scheduling problem has been discussed in several previous work [68, 69, 70, 71]. In [68], the authors first proposed a node/arc formulation, which is based on multi-flow problem with additional scheduling constraints. The formulation generates

a large number of constraints, which makes the optimal result too expensive for a large topology. The authors also proposed a continuous relaxation of the original formulation where the column generation [72] method could be used to obtain the optimal results. According to the reported result, the resolution time for obtaining the optimal result is very large even for small scale network topologies. Obtaining the optimal solution for the joint routing and scheduling problem is not practical for a real network with dynamic traffic requirement.

Heuristic routing algorithms for minimum length schedules have been discussed in [69] [73]. Both papers assumed an existing heuristic scheduling algorithm and proposed iterative routing algorithms that aim to minimize the length of the schedule. The time complexity of the above routing algorithms is determined by the number of iterations to perform.

2.3.4 Path Selection in MMR Networks

Similar to WiMAX mesh networks, in MMR networks, the base station (BS) acts as the Internet gateway and is responsible for constructing the downlink and uplink routing tree [62]. Figure 2.2 shows an example downlink routing tree. Lines in the figure represent the links for downlink traffic.

When roaming around the network, the mobile station should perform handover to maintain connectivity and a good QoS. The most important task of handover is to select an appropriate path between the mobile station and the base station. Algorithms for establishing routes between relay stations and the base station are

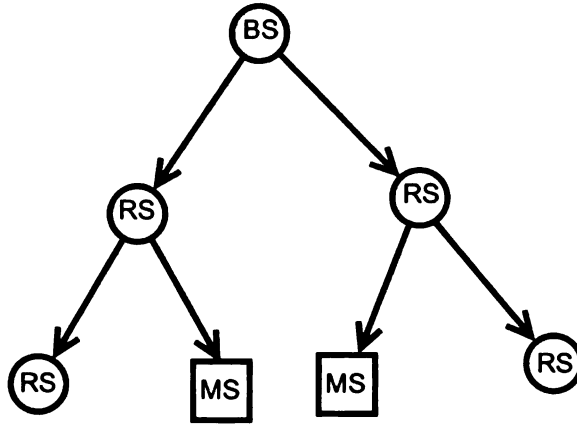


Figure 2.2. A Downlink Routing Tree Example in MMR Networks

discussed in several contributed documents [74, 75, 76, 77] to the IEEE 802.16's relay task group. Once the routing tree consisting of the base station and relay stations are constructed, the path selection problem for mobile stations is equivalent to the access station (AS) selection problem. The access station is the parent node of the mobile station in the downlink routing tree and it could be either a base station or a relay station. We denote the access link as the link between the mobile station and its access station. Note that the potential access stations of a mobile station can be in different cells.

Relay networking has been a research topic in wireless cellular networks. Sreng, et al. [78] investigated the relay node selection strategies in cellular networks with peer-to-peer relaying. The authors proposed relay node selection strategies based on distance and pathloss for two-hop relaying networks. The distance-based selection metrics include shortest total distance, least longest hop and shortest relay hop distance. The pathloss-based selection metrics include minimum total pathloss, least maximum pathloss and minimum relaying hop pathloss. Distance-based selection

strategies are simpler to carry out, however are less effective than pathloss-based strategies due to the fact that the radio signal strength is affected by shadowing and multi-path fading effects in addition to distance.

The path selection problem for mobile stations has been discussed in several contributed documents to the IEEE 802.16's relay task group. Oyman, et al. [79] proposed two end-to-end path selection metrics for mobile stations: a capacity-based metric and a throughput-based metric. The capacity-based metric computes the harmonic mean of the capacities over individual links. The capacity of each link is calculated based on the Signal to Interference and Noise Ratio (SINR). The throughput-based metric computes the harmonic mean of maximum throughputs over individual wireless links. Per-link throughput estimation accounts for link errors, modulation and coding schemes.

CHAPTER 3

QoS-Aware Fair Rate Allocation in CSMA/CA Based Wireless Mesh Networks

In this chapter we focus on the problem of providing QoS support for real-time flows, while allocating bandwidth to elastic flows fairly. Previous work [54, 80, 46, 47, 48, 50, 49, 55, 81, 57, 56] normally consider these two problems separately by focusing either on admission control of real-time flows or rate allocation of elastic flows. These approaches cannot be applied directly in a wireless mesh network that contains both real-time and elastic flows. The major contribution of this chapter is the proposed QUOTA framework providing both QoS support and fair, efficient rate allocation for wireless mesh networks.

Our framework is different from the work mentioned above in three very important

aspects. First, we consider real-time flow QoS support in addition to rate allocation for elastic flows. Higher priority is given to real-time flows by reserving bandwidth for them while utility maximization based rate allocation is used as a tool to allocate the left-over bandwidth among elastic flows fairly. Second, rate allocation is performed at the Internet gateway instead of relying on distributed mesh routers. To support the QoS of real-time flows in mesh networks, a centralized method is more suitable than a distributed one because it provides more accurate bandwidth reservation for real-time flows and faster, more stable rate allocation for elastic flows. This is vital for real-time flows since they are very sensitive to network conditions and normally last for a short period of time. In fact, the IEEE 802.16 (WiMAX) mesh network [7] provides a centralized mechanism for rate allocation and link scheduling. More detailed discussion on the centralized rate allocation will be given in Section 3.2 when we introduce the implementation details of the QUOTA framework. Third, we employ an adaptive method for accurately estimating the capacity of the network, which is the basis for efficient bandwidth reservation and rate allocation.

3.1 Theoretical Framework

In this chapter, we focus on the wireless mesh backbone formed by several mesh routers. We consider the scenario where all the mesh routers are deployed by a single organization, for example, an Internet Service Provider (ISP). In this case, all the mesh routers are configured and controlled by the organization. For the ease of presentation, we consider a scenario shown in Figure 3.1, which is a branch of a

typical wireless mesh network. Different branches can be configured to use orthogonal channels to eliminate interference between them. This scenario is similar to the one that was studied in [12]. In this chapter, we assume that wireless links are bidirectional. In addition we do not consider adaptive link capacity, so we assume the capacities of all links are equal. Note that similar link capacity assumptions are made in [47, 49]. Our model can be easily modified to take different link capacities into consideration, such as in [46].

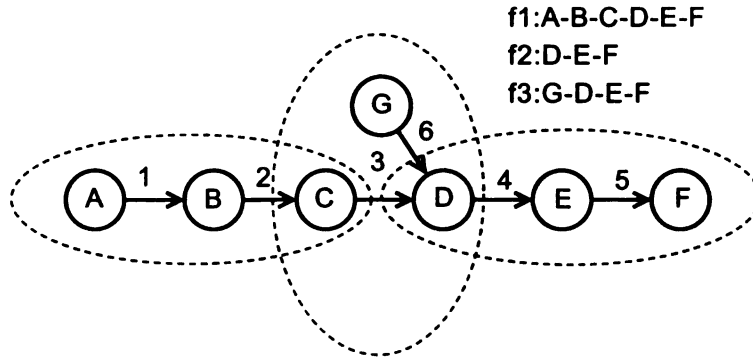


Figure 3.1. A Branch of the Mesh Network

In the mesh branch shown in Figure 3.1, node F is the Internet gateway. We denote the set of *network flows* (end-to-end application flows) as F . Also we denote the set of *link flows* (flows between directly connected nodes) as L . Every network flow $f \in F$ consists of one or more link flows. Every link flow $l \in L$ carries at least one network flow. The legend of Figure 3.1 shows three network flows in the mesh branch.

3.1.1 Link Contention Graph and Rate Allocation Feasibility

In an IEEE 802.11 MAC based wireless network, two link flows contend for channel access if the source or destination of one link flow is within the interference range of the source or destination of another one [47, 49]. We can define a *link contention graph* $G(V, E)$ based on the contention relationship between different link flows. The vertex set V contains all the link flows in the network. An edge in set E indicates that two vertices (two link flows) contend with each other. Figure 3.2 shows the contention graph of the mesh branch shown in Figure 3.1.

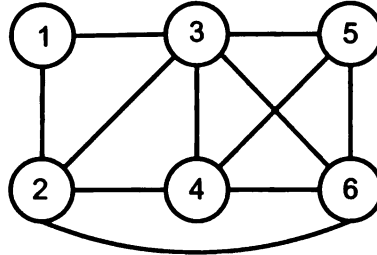


Figure 3.2. Link Contention Graph of the Mesh Network Scenario Shown in Figure 3.1

The link contention graph captures the interference among different link flows. An important concept associated with the link contention graph is the *maximal clique*. A set of nodes is a clique if its induced subgraph is complete. A maximal clique is defined as a clique that is not contained in any other clique [82]. We denote the set of maximal cliques of a contention graph as C . The maximal cliques can be obtained from the link contention graph using the Bierston algorithm [82]. In the link contention graph given in Figure 3.2, we have three maximal cliques, $C_1 = \{1, 2, 3\}$, $C_2 = \{2, 3, 4, 6\}$ and $C_3 = \{3, 4, 5, 6\}$. In fact, a maximal clique represents a contention region in which all link flows interfere with each other. The dotted ellipses shown in Figure 3.1

illustrate contention regions corresponding to the above maximal cliques. A maximal clique can also be considered as a limited resource shared and contended by different link flows within it. For any maximal clique, at any time, only a single link flow can be active. The above constraint is denoted as the *clique constraint*. In the rest of this chapter, we will simply use clique to refer to maximal clique.

In order to formally state the clique constraint, let $\mathbf{x} = (x_f, f \in F)$ denote the *rate allocation vector* of network flows; let $\mathbf{y} = (y_l, l \in L)$ denote the rate allocation vector of link flows derived from \mathbf{x} . Also let b_l denote the link capacity for link flow l when it is active in isolation. Rate vector \mathbf{x} and \mathbf{y} are said to be *scheduling feasible* if they can be scheduled under the clique constraint. The scheduling protocols considered here are Medium Access Control (MAC) layer protocols. In this chapter, we use the IEEE 802.11 MAC as the underlying scheduling protocol.

We denote the necessary and sufficient condition that guarantees scheduling feasibility as the *scheduling feasibility constraint*. In order to derive this constraint, let us first define a matrix $\mathbf{R} = \{R_{lf}\}$:

$$R_{lf} = \begin{cases} 1, & \text{if link flow } l \text{ carries network flow } f \\ 0, & \text{otherwise} \end{cases}$$

Also define matrix $\mathbf{Q} = \{Q_{nl}\}$:

$$Q_{nl} = \begin{cases} 1, & \text{if clique } n \text{ contains link flow } l \\ 0, & \text{otherwise} \end{cases}$$

As an example, for the mesh scenario shown in Figure 3.1 and the associated link

contention graph shown in Figure 3.2, we have

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

and

$$\mathbf{Q} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

It has been shown that [46, 47], for a perfect contention graph [83], the scheduling feasibility constraint can be written as

$$\sum_{l:Q_{nl}=1} \frac{y_l}{b_l} \leq 1, \quad \forall n \in C$$

where

$$y_l = \sum_{f:R_{lf}=1} x_f, \quad \forall l \in L$$

In fact, the constraint states that the sum of the normalized rate of all link flows in the clique, cannot exceed the *normalized clique capacity*, which is 1 [47].

However, it is difficult and expensive to check if the link contention graph is perfect. It has been shown that [47], for a general graph, the scheduling feasibility constraint can be written as

$$\sum_{l:Q_{nl}=1} \frac{y_l}{b_l} \leq \frac{2}{3}, \quad \forall n \in C$$

by reducing the normalized clique capacity to $2/3$.

Based on our link capacity assumption given at the beginning of this section, we denote the capacity of all wireless links to be b . So we can write the scheduling feasibility constraint in another form

$$\sum_{l:Q_{nl}=1} y_l \leq \frac{2}{3}b, \quad \forall n \in C \quad (3.1)$$

where the right-hand side of the inequality is denoted as the *clique capacity*.

Scheduling feasibility does not guarantee that the rate vector is *throughput feasible*, which means the throughput of a flow equals to the allocated rate. This is due to the inefficiency of the underlying scheduling protocols. For example, due to the random nature of the IEEE 802.11 MAC, some idle time is wasted during the backoff period. We introduce the *effective clique capacity* τ_n for every clique $n \in C$ such that the constraint

$$\sum_{l:Q_{nl}=1} y_l \leq \tau_n, \quad \forall n \in C$$

guarantees the throughput feasibility of the rate vector. We denote the above constraint as the *throughput feasibility constraint*. To simplify the representation of the throughput feasibility constraint, let us define the *clique-flow matrix* $\mathbf{P} = \mathbf{Q}\mathbf{R}$ ($P_{nf} = Q_{nl}R_{lf}$). In fact P_{nf} represents the number of link flows in clique n that carries network flow f . As an example, for the mesh scenario shown in Figure 3.1 and the associated link contention graph shown in Figure 3.2, we have

$$\mathbf{P} = \begin{bmatrix} 3 & 0 & 0 \\ 3 & 1 & 2 \\ 3 & 2 & 3 \end{bmatrix}$$

Denote τ as the vector of effective clique capacities. Then the throughput feasibility constraint can be written as

$$Px < \tau \quad (3.2)$$

The effective capacity of a clique depends on several factors, such as the underlying scheduling protocol, the number of competing link flows in this clique and the location of link flows [48]. It is difficult to determine the effective capacity of a clique in advance. In our framework, the effective capacity of a clique has the maximum value of $(2/3)b$ and is adaptively estimated according to the network conditions.

3.1.2 Admission Control

Assume that a network planner has a global view of the contention graph and the traffic pattern, and the network flow set F is divided into real-time flow set F^r and elastic flow set F^e . We give a higher priority to real-time flows over elastic flows by performing admission control and bandwidth reservation first and allocate the left-over bandwidth to elastic flows afterwards. We sequentially process all real-time flows. The order of processing will be discussed in Section 3.2.2. For a real-time flow $f \in F^r$, it is admitted if

$$P_{nf}r_f < \tau'_n, \quad \forall n \in C$$

Otherwise, it is rejected. P_{nf} denotes the number of link flows in clique n that carries network flow f ; r_f is the rate requested by real-time flow f . So $P_{nf}r_f$ represents the bandwidth used by flow f in clique n . τ'_n is the available bandwidth of clique n , which is set to the effective capacity of the clique at the beginning of the admission

control process. If real-time flow f is admitted, which means its requested rate can be supported by all cliques, the rate allocation x_f is equal to r_f ; otherwise, $x_f = 0$. In addition, the available bandwidth of every clique n is updated as $\tau'_n = \tau'_n - P_{nf}x_f$. The bandwidth reservation of real-time flows will be discussed in Section 3.1.3.

The above admission control strategy gives absolute priority to real-time flows. However, other admission control strategies can be adopted according to the policy of the mesh network operator. For example, before admission control, a certain proportion of the available bandwidth of every clique can be reserved for elastic flows to prevent their starvation.

3.1.3 Utility Maximization Based Rate Allocation

After the admission control is completed for real-time flows, we may allocate bandwidth to elastic flows. In order to guarantee fairness among different elastic flows, we use a well-developed utility maximization framework. This framework was originally proposed for Internet congestion control [44, 45] and recently has been used for fair rate allocation in wireless networks [46, 47, 48, 49].

We associate a utility function $U_f(x_f)$ to every elastic flow $f \in F^e$ [10]. Further, we assume that this function is increasing, strictly concave and twice continuously differentiable [46, 47, 49]. It has been shown that [44, 51], by maximizing $\sum_{f \in F^e} U_f(x_f)$, a certain targeted fairness is achieved among elastic flows. Furthermore, if we choose the utility function $U_f(x_f) = \ln(x_f)$ for every elastic flow, proportional fairness is

achieved [44, 51]. More formally, the objective can be written as

$$\mathbf{P}: \max_{x_f \geq 0} \sum_{f \in F^e} U_f(x_f), \text{ subject to } \mathbf{P}\mathbf{x} \leq \boldsymbol{\tau} \quad (3.3)$$

The above problem is referred to as the system primal problem. It is a typical convex optimization problem [84], which can be solved by using the Lagrange duality [44, 45].

The dual problem is defined as

$$\mathbf{D}: \min_{\boldsymbol{\lambda} \geq 0} D(\boldsymbol{\lambda}) \quad (3.4)$$

where

$$\begin{aligned} D(\boldsymbol{\lambda}) &= \max_{x_f \geq 0} \left(\sum_{f \in F^e} U_f(x_f) - \boldsymbol{\lambda}^T (\mathbf{P}\mathbf{x} - \boldsymbol{\tau}) \right) \\ &= \sum_{f \in F^e} \max_{x_f \geq 0} (U_f(x_f) - x_f \sum_{n \in C} \lambda_n P_{nf}) \\ &\quad + \sum_{n \in C} \lambda_n \tau_n - \sum_{f \in F^r} x_f \sum_{n \in C} \lambda_n P_{nf} \end{aligned}$$

Let us define

$$\mu_f = \sum_{n \in C} \lambda_n P_{nf}$$

In fact, the Lagrange multiplier λ_n can be interpreted as the shadow price [44] of clique n , which is the cost of a unit flow accessing the channel in clique n [49]. Also $x_f \mu_f$ may be interpreted as the total price charged for flow f for transmitting at rate x_f .

To solve the dual problem, we use an iterative algorithm with the help of the gradient projection method [49]. The algorithm involves all network flows and maximal cliques and is given an initial rate allocation vector \mathbf{x} . For real-time flow f , x_f is determined by the admission control process described in Section 3.1.2 and does not

change during the iterative process. For elastic flows, initial rate can be randomly chosen.

At every iteration, for every clique n , it receives the rate information of all flows f where $P_{nf} \neq 0$ and updates the shadow price λ_n using

$$\lambda_n(t+1) = [\lambda_n(t) - \gamma \frac{\partial D(\lambda(t))}{\partial \lambda_n}]^+$$

where γ is the step size and t is the iteration number. Since

$$\frac{\partial D(\lambda)}{\partial \lambda_n} = \tau_n - \sum_{f \in F^e} x_f P_{nf} - \sum_{f \in F^r} x_f P_{nf}$$

we have

$$\lambda_n(t+1) = [\lambda_n(t) - \gamma(\tau_n - \sum_{f \in F^e} x_f(t) P_{nf} - \sum_{f \in F^r} x_f P_{nf})]^+ \quad (3.5)$$

The above equation reflects the law of supply and demand [47, 49]. When the rate demand exceeds (less than) the effective capacity of clique n , the cost of a unit flow accessing the channel in clique n increases (decreases). The above equation also shows the fact that every clique n reserves necessary bandwidth, $\sum_{f \in F^r} x_f P_{nf}$, for real-time flows. Note that the rates of real-time flows do not change during the iteration, so the bandwidth reservation will not be affected by the rate allocation of the elastic flows. After the price update, the clique shadow price is reported to the source of all elastic network flows f where $P_{nf} \neq 0$.

For every elastic flow f , the source of this flow receives shadow price information from all maximal cliques n where $P_{nf} \neq 0$. Then a new transmission rate of the flow is obtained by solving the following problem

$$\max_{x_f \geq 0} (U_f(x_f) - x_f \mu_f) \quad (3.6)$$

which can be interpreted as maximizing the net utility (utility minus cost). Since function $U_f(x_f)$ is strictly concave, the unique solution of the above problem exists and can be expressed as

$$x_f^* = U_f'^{-1}(\mu_f)$$

Since $U_f(x_f) = \ln(x_f)$, the maximizer $x_f^* = \frac{1}{\mu_f}$. The new rate information of flow f is reported to all maximal cliques n where $P_{nf} \neq 0$ and will be used in the next iteration.

It has been shown that [46, 49], by choosing the appropriate step size γ , starting from any initial rate vector \mathbf{x} , the above iterative algorithm will converge to the optimal solution $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$, and the solution is primal-dual optimal, which means \mathbf{x}^* is also the optimal rate vector for the primal problem.

The above discussion introduces the general idea of our framework. However, in order to use this framework in a wireless mesh network, there are many practical issues that need to be considered. First, the construction of the contention graph and its maximal cliques is a challenging problem. Second, it is a difficult task to perform clique price update since the clique is a virtual concept. Third, the communication between the sources of network flows and the cliques should be reliable enough to guarantee the correct execution of the algorithm. Last, in order to guarantee the throughput feasibility and the QoS of real-time flows, we need an efficient on-line algorithm to estimate the effective clique capacities according to the network conditions. In the next section, we introduce the implementation of the QUOTA framework by solving the above-mentioned practical issues.

3.2 Framework Implementation and Practical Issues

The implementation of the framework executes in a round-by-round fashion. Every round has a fixed length duration consisting of two phases: distributed clique construction and centralized admission control and rate allocation. The detailed description of these two phases and the associated control messages are given in this section.

It is reasonable to assume that all branches share a common control channel that is orthogonal to the channels used for data transmission. The control channel can be used for mesh backbone management. It will also be used by QUOTA for control message exchange. The multi-interface and multi-channel mesh network architecture has been proposed and investigated in both academia and industry [27, 85, 86, 25, 87, 12] and becomes essential for the wide deployment of the wireless mesh networks. The use of a dedicated control channel can improve the reliability of the framework and leave more bandwidth for data transmission. When the control channel for the mesh backbone is not available, QUOTA control messages have to compete with data packet transmissions, which results in potential control message loss and less bandwidth for data transmission. In this case, a high priority queue can be used for control messages when packet transmissions are scheduled. We will compare the performance of QUOTA with and without a dedicated control channel in Section 3.3.

3.2.1 Distributed Clique Construction

The framework presented in Section 3.1 requires the construction of a global link contention graph and its maximal cliques. The Bierstone algorithm used for the maximal clique problem has exponential time complexity [88]. In our implementation, we use a distributed maximal clique construction method similar to [89, 47, 49] through which we obtain the maximal cliques without the construction of a global link contention graph. Since every mesh router has limited number of neighbors, even with exponential time complexity, to obtain the maximum clique set of the local link contention graph is fast.

Every link flow in the network has an associated local link contention graph that contains itself and all other link flows that interfere with it. Local link contention graphs are subgraphs of the global link contention graph and they are usually overlapping. We let individual mesh routers construct the local link contention graphs for all link flows that originate from it. The mesh router then applies the Bierstone algorithm [82] on the contention graphs to obtain a set of maximal cliques. It has been shown that, the collection of maximal cliques of different local link contention graphs gives us the complete set of maximal cliques of the global link contention graph [49].

In order to construct the local link contention graph of a given link flow, the mesh router needs to know the information about all other link flows that are in the interference range of the sender or the destination of this flow. In our implementation, the interference range is set to be identical to the transmission range so the mesh router can construct the local link contention graph via control message exchange.

We use two types of messages: *beacon messages* and *link messages*. Similar types of messages are used in [89, 47]. A beacon message is associated with a link flow and contains the source and destination identifiers of this link flow. Similarly a link message is associated with a link flow and contains the information of itself and all other flows that interfere with it. Every mesh router periodically broadcasts beacon messages and link messages that are associated with all the link flows originate from it or are destined to it. Link messages are constructed based on the information obtained from beacon messages so they are sent less frequently than beacon messages. In order to see why we need two types of messages and how these messages are used, let us consider a simple example shown in Figure 3.3. A similar example is given in [89]. Dotted circles in the figure shows the transmission range of node A and node C. Assume node A needs to construct the local link contention graph of link flow 1. Link flow 2 should be included in the contention graph since node B is in the transmission range of node C. By receiving beacon messages from node C, node B knows that flow 2 interferes with flow 1. However, node A cannot receive beacon messages from node C since it is out of the transmission range of node C. By receiving link messages from node B, node A then knows the information about link flow 2 and includes it in the contention graph of link flow 1.

At the end of the clique construction phase, each mesh router sends *clique messages* to the gateway node. Each clique message is associated with a link flow that originates from the mesh router and contains three parts: the source and destination of this flow; all the maximal cliques to which this link flow belongs; network flows that this link flow carries. If the mesh router is the source of a real-time network

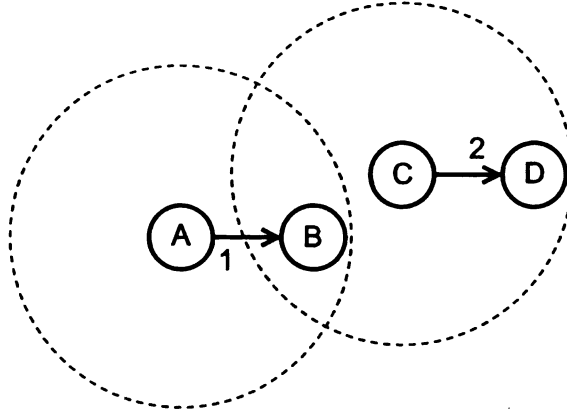


Figure 3.3. Local Link Contention Graph Construction Example

flow, the rate and delay requirements of this flow are also included. Note that elastic flows do not have specific rate and delay requirements. By receiving all these clique messages, the gateway node is able to construct the set of maximal cliques (denoted as C) of the global link contention graph, and the set of network flows (denoted as F). In addition, based on the information of network flows contained in the clique messages, it can construct the clique-flow matrix P .

Figure 3.4 shows an example of the clique construction process. The corresponding network topology and traffic pattern are shown in Figure 3.1. The associated global link contention graph is shown in Figure 3.2. In this scenario, each mesh router only has one local link flow that originates from it. The row “local contention graph” of the table shows the local contention graphs constructed at different mesh routers. The row “maximal cliques” shows the maximal clique decomposition at different mesh routers.

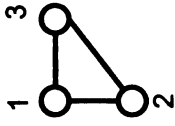
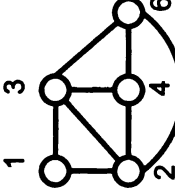
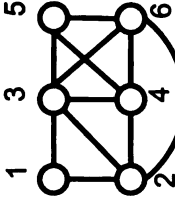
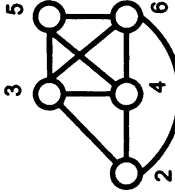
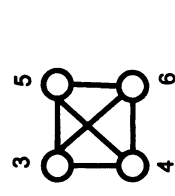
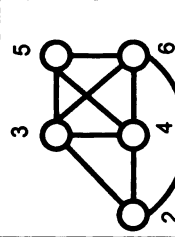
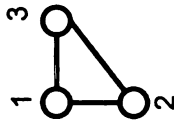
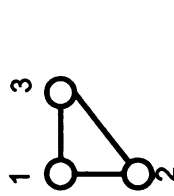
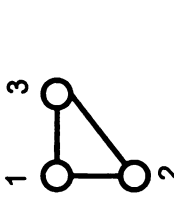
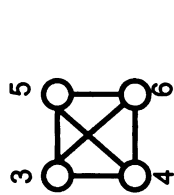
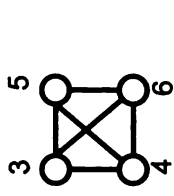
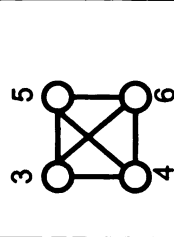
	node A (link flow 1)	node B (link flow 2)	node C (link flow 3)	node D (link flow 4)	node E (link flow 5)	node G (link flow 6)
local contention graph						
maximal cliques						

Figure 3.4. Distributed Clique Construction Example. The Associated Network Topology is Shown in Figure 3.1; The Associated Global Link Contention Graph is Shown in Figure 3.2.

3.2.2 Centralized Admission Control and Rate Allocation

Based on the information collected during the clique construction phase, the gateway node performs admission control and rate allocation using the algorithm described in Section 3.1. In fact, the gateway node can be considered as a simulator that executes the above-mentioned algorithm. The input to the simulator consists of the maximal clique set C , network flow set F , the clique-flow matrix \mathbf{P} and the QoS requirements of real-time flows. The output of the simulator is the converged rate allocation vector. For a real-time flow, if it is rejected, the rate assignment will be zero; otherwise, the rate assignment is the rate requested by the source. In order to reduce the probability that admitted real-time flows get rejected in later rounds, during the admission control process described in Section 3.1.2, we give higher priority to real-time flows admitted in previous rounds by processing their rate requests before newly arrived real-time flows.

When the rate allocation vector is available, the gateway node sends *rate messages* to the source of each network flow. Rate messages contain the assigned rate of the associated network flow. By receiving the rate message, the source of a network flow knows the rate to use for the next round.

As explained at the beginning of this chapter, the motivation of relying on the gateway node instead of several mesh routers to perform the admission control and rate allocation is for better QoS support of real-time flows. A distributed implementation [47, 49] would require several rounds to perform the iterative process described in Section 3.1.3 and to find the correct effective clique capacities described in Section

3.2.3. Each round of the distributed implementation corresponds to one iteration of the rate allocation algorithm and it involves information exchange about flow rate and clique price among wireless routers. The long process of finding the optimal rate allocation and determining correct effective clique capacities, which can last for tens of seconds [49], has a detrimental effect on the QoS of real-time flows. In contrast, executing the admission control and rate allocation algorithm virtually at the gateway node yields an optimal rate allocation in a single round and determines the correct effective clique capacities in a few rounds. We will show the benefits of using a centralized rate allocation through simulation results in Section 3.3.

The utility maximization based rate allocation problem is a convex optimization problem. It has been pointed out that using a centralized algorithm, the optimal solution can be obtained in worst-case polynomial-time complexity [90, 84]. Therefore our centralized implementation for the rate allocation phase is scalable.

3.2.3 Adaptive Effective Clique Capacity

As we mentioned in Section 3.1, the effective clique capacity cannot be determined in advance. In our implementation, we use a measurement-based method to dynamically estimate the effective capacities of maximal cliques. When there are no real-time flows in the network, we use a conservative value $0.6b$ as the clique capacity. This value is slightly smaller than $(2/3)b$ that guarantees scheduling feasibility as shown in Equation 3.1.

When there are real-time flows in the network, the receiver of any real-time flow

measures the average delay of this flow during the clique construction phase and this information is attached to the clique messages sent to the gateway. The effective clique capacity adjustment is performed before admission control and rate allocation. If the maximal clique set and the traffic pattern are the same as the last round, the initial effective capacity of every maximal clique is set to the value determined in the last round. Then for every real-time flow f , the gateway node checks that if the average measured delay exceeds the delay requirement. If yes, for every clique n , the effective clique capacity τ_n will be reduced by $P_n f a$, where a is a small constant. This is equivalent to reserving extra bandwidth $P_n f a$ for the real-time flow f in clique n as shown in Equation 3.5. If the maximal clique set or the traffic pattern changes in a new round, the effective capacity of every maximal clique will be set to $0.6b$ and no further capacity adjustment will be performed in this round.

The major purpose of dynamically adjusting the effective clique capacities is to meet the rate and delay requirements of the real-time flows by guaranteeing throughput feasibility. By reserving extra bandwidth for real-time flows, we expect to reduce the delay and at the same time meeting its rate requirement. However, when there are only elastic flows in the network, we use a pre-determined conservative value as the effective clique capacity, which may not guarantee throughput feasibility.

3.3 Performance Evaluation

In this section, we present simulation results for the QUOTA framework in mesh networks consisting of real-time and elastic flows. Also we compare the performance

of QUOTA with SWAN, which is introduced in Section 2.3.1. We choose SWAN since its QoS support for real-time flows is based on the rate control of the elastic flows, which is very similar to QUOTA. The evaluation metrics include the delay and loss rate of real-time flows, the aggregate throughput of the network and the fairness index of elastic flows. Given a rate vector \mathbf{x} of size n , the fairness index [91] is defined as

$$F(\mathbf{x}) = \frac{(\sum x_i)^2}{n(\sum x_i^2)}$$

3.3.1 Simulation Setup

We use ns-2 [92] (version 2.29) for our simulation. For the QUOTA framework, the duration of each round is set to be 4 seconds. In the distributed clique construction phase, 2 seconds are used for beacon message and link message exchange. Each mesh router periodically sends beacon messages and link messages. The inter-message intervals of beacon and link messages are 0.2 and 0.9 seconds, respectively. Another one second is used for sending clique messages to the Internet gateway. After admission control and rate allocation, the gateway node sends rate messages to the source of each network flow. If the source fails to get the rate message during a round, it will continue using the rate assigned at the last round. Some part of the QUOTA implementation in ns-2 is based on the code used in [47]. The implementation code of the SWAN framework is obtained from the authors' website.

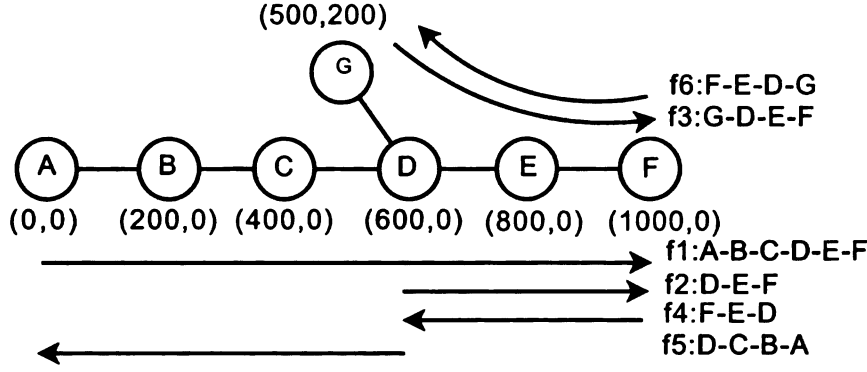


Figure 3.5. A Single-Branch Scenario

3.3.2 Simulation Results of a Single-branch Scenario

The simulated single-branch scenario is shown in Figure 3.5. All nodes use two 2Mbps 802.11 radios tuned to orthogonal channels used by data transmission and control message exchange. The transmission range and interference range are set to be 250m. The coordinates of nodes are given in the figure. In this scenario, we have six network flows. The routes for these flows are manually set and are shown in the figure. Flows f_1 , f_2 , f_4 and f_5 are elastic flows with constant bitrate (CBR) traffic source. The packet size is set to be 1000 bytes. The sending rate of the traffic source is adjusted at the application layer under QUOTA framework while SWAN adjusts the source rate using a traffic shaper between and IP and MAC layer. Note that both QUOTA and SWAN can work with a TCP-based traffic source. However, in this chapter we consider TCP as a form of rate allocation for elastic flows [44, 49] and use it as a baseline for comparison. Elastic flows last for the whole simulation period, which is 400 seconds. Real-time flows f_3 and f_6 are from voice applications simulated as CBR traffic sources with 32kbps rate requirement and 50ms maximal tolerable one-way delay. The packet size is set to be 80 bytes. Real-time flows start

at 100 seconds and end at 300 seconds.

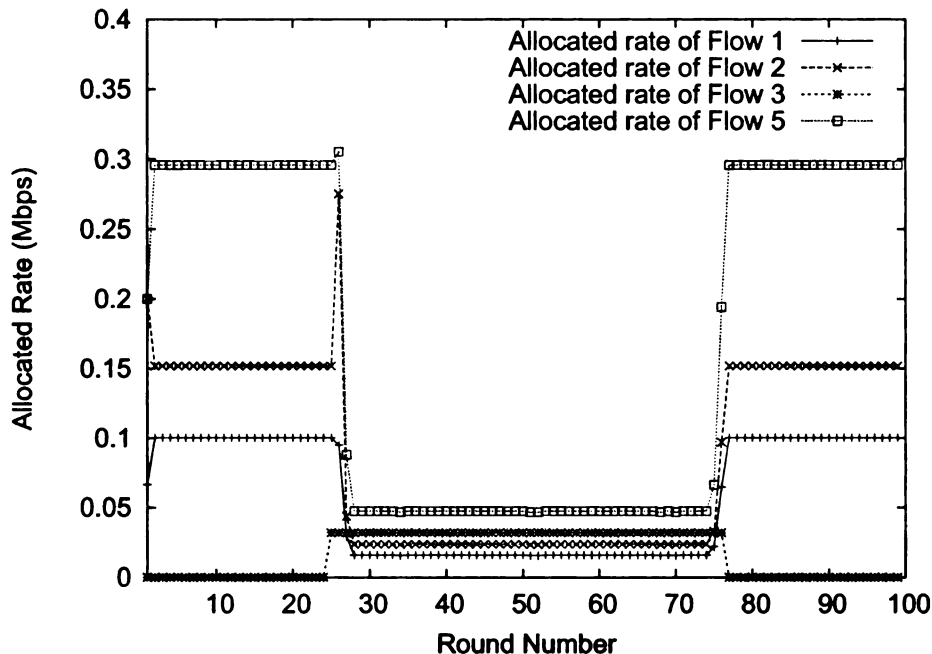


Figure 3.6. Allocated Rate of Different Flows at Different Rounds under QUOTA

Figure 3.6 shows the rate allocation at different rounds under the QUOTA framework. The rate allocations of flow f_4 and flow f_6 are not shown in the figure since they are the same as flow f_2 and flow f_3 respectively. During $[0,100]$ seconds, there are no real-time flows in the network and the available bandwidth is allocated to elastic flows. At 100 seconds, two real-time flows enter the network and the rate allocation is quickly adjusted to meet the QoS requirements of them. When real-time flows leave the network at 300 seconds, the available bandwidth for elastic flows increases and the new rate allocation is the same as that in the period $[0,100]$. Figure 3.7 shows the throughput of flows at different rounds under the QUOTA framework. The throughput of flow f_4 and flow f_6 are not shown in the figure since they are similar to flow f_2 and flow f_3 respectively. We observe that before real-time flows enter the network,

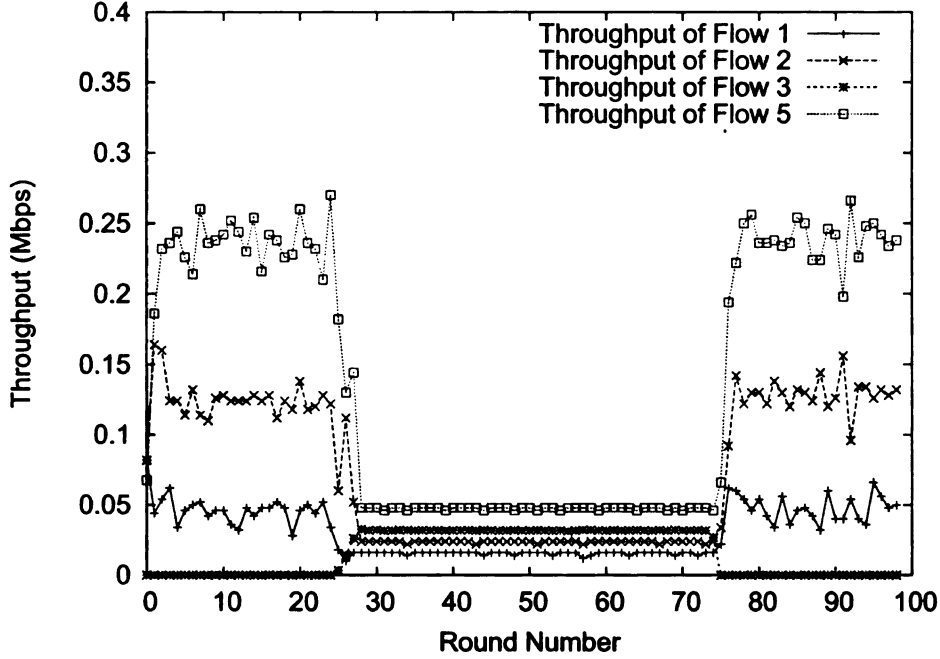


Figure 3.7. Throughput of Different Flows at Different Rounds under QUOTA

the rate allocation is not throughput feasible. Starting at 100 seconds, the effective clique capacities are adjusted according to the delay requirements of real-time flows. After a few rounds, the rate requirements of real-time flows are satisfied and the loss rates of real-time flows are under 1% during the period [100,300]. The rate allocation of elastic flows also becomes throughput feasible during this period. Figure 3.8 shows the average delay of real-time flows in logarithmic scale at different rounds under the QUOTA framework. When real-time flows enter the network, the delay requirements are not satisfied. After a few rounds of clique capacity adjustment, the delays of real-time flows are under the maximal tolerable value and are stably maintained during the whole session.

We also simulate SWAN and TCP in the single-branch scenario. For SWAN, we tuned the maximum sending rate of elastic flows and the delay threshold to make sure

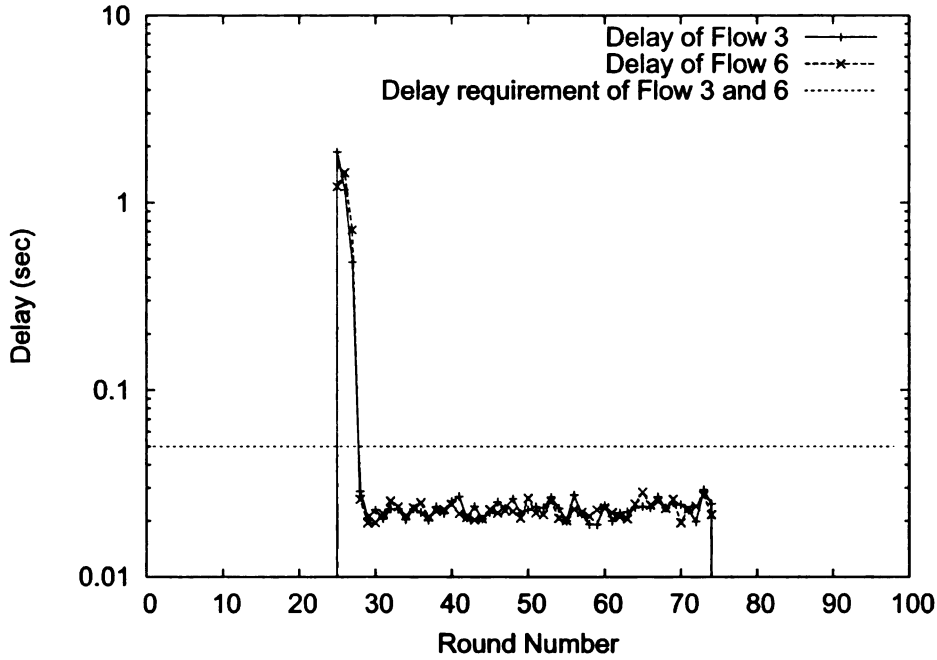


Figure 3.8. Average Delay of Two Real-time Flows at Different Rounds under QUOTA

that real-time flows are admitted and their delay and rate requirements are successfully met. When TCP is used for elastic flows, both rate and delay requirements of real-time flows are violated, which shows the necessity of using admission control and bandwidth reservation to support the service requirements of real-time flows. Table 3.1 shows the aggregate throughput of all flows, the average delay of real-time flows and the fairness index of elastic flows under QUOTA, SWAN and TCP. The aggregate throughput under QUOTA is higher than SWAN and slightly smaller than TCP. In addition, the fairness index of elastic flows under QUOTA is greatly improved over SWAN and TCP. In previous work on fair rate allocation for elastic flows, utility maximization based framework usually achieves higher aggregate throughput and better fairness than TCP [48]. However, QUOTA supports QoS of real-time flows by reducing the available bandwidth for elastic flows, which may result in a smaller ag-

aggregate throughput than TCP as seen in this scenario. QUOTA outperforms SWAN in two aspects: First, to support the same level of QoS for real-time flows, QUOTA maintains more bandwidth for elastic flows than SWAN. Second, QUOTA allocate bandwidth more fairly than SWAN.

Table 3.1. Performance Comparison of the Single-branch Scenario (CBR Real-time Traffic)

	TCP	SWAN	QUOTA
Aggregate Throughput (kbps)	406	297	385
Average Delay of Real-time Flows (ms)	431	49	46
Fairness Index	0.62	0.46	0.81

Here are some reasons for the performance gap between SWAN and QUOTA. First, SWAN performs rate control of elastic flows on a per-node basis by using a traffic shaper between the IP and MAC layer; QUOTA directly adjusts the sending rate of elastic flows at the application layer, which provides more accurate per-flow rate adjustment. Second, the delay threshold value used in SWAN, which triggers the rate control of elastic flows, only reflects the contention level of a single maximal clique (contention region). For networks with multi-hop flows spanning several maximal cliques, it is difficult to choose an appropriate delay threshold according to the delay requirement of the real-time flow. By using the link contention graph and maximal clique decomposition, QUOTA's bandwidth reservation for real-time flow is more flexible and efficient. The rate allocation of elastic flows is fairer with the help of the utility maximization framework.

The voice traffic can also be modeled as an ON/OFF source with exponential distributed ON and OFF period [93]. We perform simulation using traffic sources with exponential ON and OFF period for real-time flows f_3 and f_6 . The average

length of ON and OFF period is 300ms. Packets are generated during ON periods at a constant bit rate of 32kbps. Other simulation parameters are the same as the previous scenario with CBR real-time traffic. Since the length of the ON or OFF period is much smaller than the interval of a round in QUOTA, we use the rate at the ON period as the rate requirements in QUOTA. This approach is conservative in the sense that it reserves bandwidth for real-time flows even they are in the OFF period. However, it is efficient in maintaining the QoS of real-time flows.

Table 3.2. Performance Comparison of the Single-branch Scenario (VBR Real-time Traffic)

	TCP	SWAN	QUOTA
Aggregate Throughput (kbps)	449	304	355
Average Delay of Real-time Flows (ms)	401	46	36
Fairness Index	0.6	0.47	0.8

Table 3.2 shows the aggregate throughput of all flows, the average delay of real-time flows and the fairness index of elastic flows under QUOTA, SWAN and TCP. Compared with simulation results of the previous scenario with CBR real-time traffic, we observe that SWAN and TCP take advantage of the additional bandwidth given by the OFF period of the real-time traffic while QUOTA does not. However, the performance benefits of QUOTA are still obvious.

3.3.3 Simulation Results of Multi-branch Scenarios

In this section, we present the simulation results of multi-branch wireless mesh network scenarios. We simulate a flat area of 600m by 600m with 40 randomly positioned stationary wireless nodes. In addition, a gateway node is placed in the center of the network. The network is divided into four branches and each branch contains one

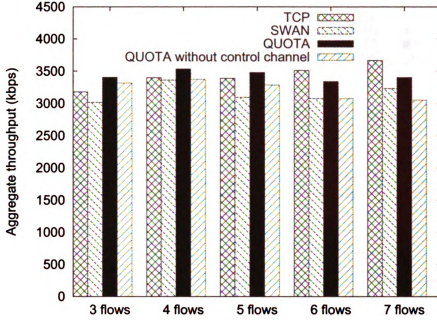


Figure 3.9. Aggregate Throughput of Network Flows

maximal clique. Different branches use orthogonal channels with 2Mbps capacity for data transmission while sharing a single control channel for QUOTA control messages. Routes between nodes are determined offline using shortest-path routing algorithms. The simulation results presented are the average values of 5 randomly generated network topologies.

In these scenarios we simulate network flows that originate from or are destined to the gateway, and we vary the number of network flows in a branch from 3 to 7 in each topology. There is one real-time flow in each branch. Real-time flows are from video applications simulated as CBR traffic sources with rate requirement 200kbps and 20ms maximal tolerable delay. The packet size is set to be 512 bytes. Elastic flows are generated using CBR traffic sources with packet size of 1000 bytes. Each simulation lasts for 400 seconds.

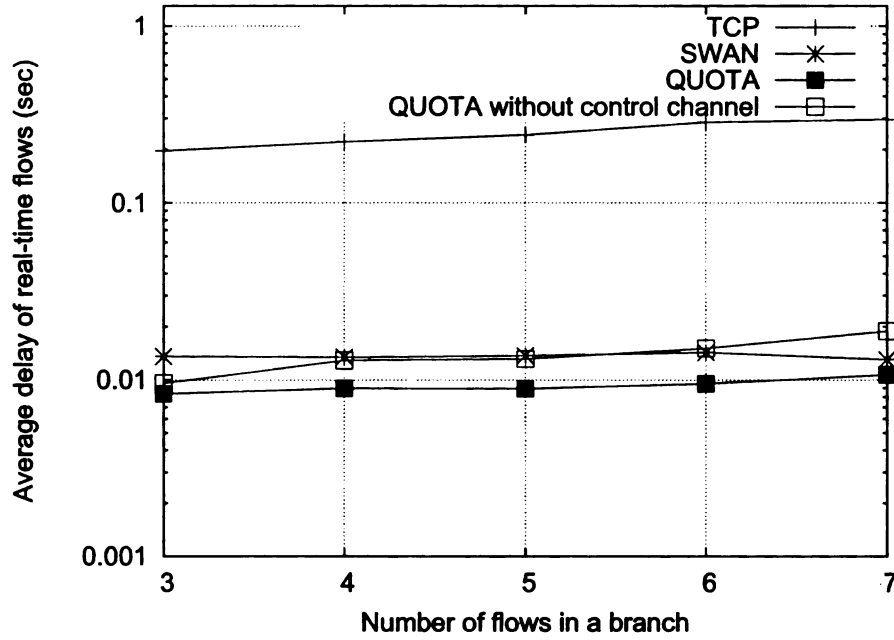


Figure 3.10. Average Delay of Real-time Flows

Figure 3.9 shows the aggregate throughput of all flows with different number of flows in each branch. The figure shows that QUOTA always achieves higher aggregate throughput than SWAN. When the number of flows in a branch is large, TCP achieves higher aggregate throughput than QUOTA and SWAN. Figure 3.10 shows the average delay of real-time flows in logarithmic scale. We observe that both the QUOTA and SWAN satisfy the delay requirements of real-time flows while TCP cannot provide the service guarantee. The average fairness index of branches are shown in Figure 3.11. We observe that QUOTA has a better fairness index than both SWAN and TCP, especially when the number of flows in each branch is large. The above simulation results indicate that, while maintaining the same level of delay for real-time flows, QUOTA allocates bandwidth to elastic flows more efficiently and more fairly than SWAN.

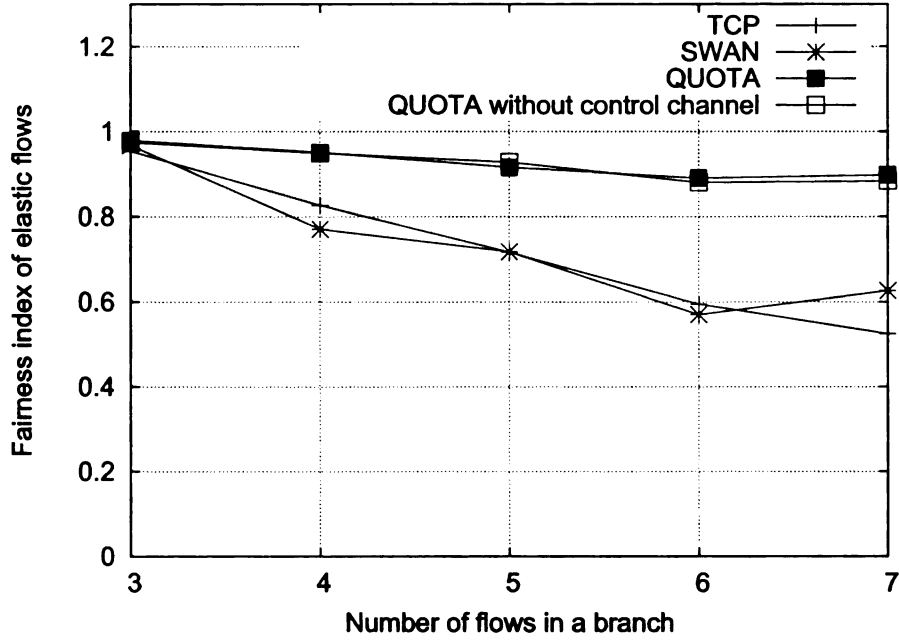


Figure 3.11. Average Fairness Index of Branches

Finally, we evaluate the overhead of the QUOTA framework by relaxing the assumption of a dedicated control control. In a new set of simulations, QUOTA control messages are sent through the respective data transmission channels of different branches. The simulation results are presented as “QUOTA without control channel” in the Figures 3.9, 3.10 and 3.11. We observe that without the dedicated control channel, QUOTA still successfully maintains the QoS of real-time flows and achieves high fairness index. Since control message transmissions compete with data packet transmissions, the aggregate throughput of QUOTA without dedicated control channel is lower than QUOTA with dedicated control channel. The normalized message overhead of QUOTA is shown in Figure 3.12. The normalized overhead is defined as the ratio between the control messages transmitted in bytes and data packets received in bytes. The message overhead increases with the increase of the number of flows in

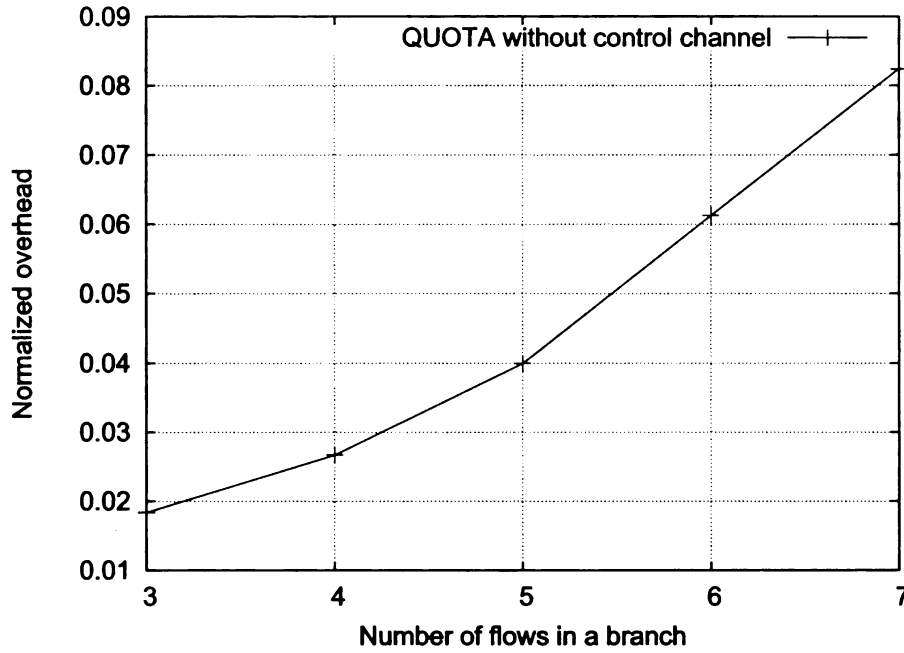


Figure 3.12. QUOTA Control Message Overhead

a branch. For a mesh branch consisting of large number of flows, using a dedicated control channel would be more appropriate. However, typical mesh network deployment limits the number of mesh routers that connect to a gateway. For example, in Google’s Muni WiFi [94, 95] mesh network deployment, small clusters of 6-8 access points are connected to a gateway using the same frequency channel. Most of the nodes are one hop away from the gateway and typical network flows are from or to the gateway. QUOTA framework operates on aggregated flows between the mesh routers and the gateway, whose number is limited by the number of mesh routers in each branch.

3.4 Summary

In this chapter we proposed a framework named QUOTA for QoS support and fair rate allocation in wireless mesh networks. Our framework uses link contention graphs and the utility maximization framework to perform admission control and rate allocation. Simulation results show that our framework successfully guarantees the QoS of real-time flows and fairly, efficiently allocates bandwidth for elastic flows in different wireless mesh network scenarios. We also compared the performance of QUOTA with the SWAN framework and explained the advantages of our framework. Both QUOTA and SWAN provide QoS support for real-time flows by controlling the rate of elastic flows. However, QUOTA outperforms SWAN in terms of the efficiency and fairness of rate allocation for elastic flows. We observed large performance gap between QUOTA and SWAN in various mesh network scenarios.

The further reduction of message overhead of QUOTA is our future work. One promising direction is to combine QUOTA with a routing protocol, such as AODV [28]. This approach has been discussed in [49].

Our framework will benefit from a more efficient underlying scheduling protocol. An improved scheduling protocol will help to increase the effective capacity of maximal cliques. In the next chapter, we will apply the rate allocation framework in wireless mesh networks with TDMA scheduling protocols.

CHAPTER 4

Optimization Based Rate

Allocation and Scheduling in

TDMA Based Wireless Mesh

Networks

Multicast is a key technology for content distribution over wireless mesh networks. However, the introduction of multicast traffic in the network should not starve the existing unicast traffic [96]. It is important to consider the coexistence of multicast and unicast traffic in the network when allocating network resources. In this chapter, we propose a framework for both rate allocation and scheduling for multicast and unicast sessions in wireless mesh networks. Our rate allocation algorithm is based on Network Utility Maximization (NUM) [52]. We first establish the notion of *trans-*

missions as a generalization of *links* and construct contention graphs based on the contention relationships among transmissions. The contention graph is used to model the rate allocation constraints. By using receiver-oriented utility functions and maximizing the total utility of the network, our rate allocation gives incentive to users to join multicast sessions, while preventing extreme unfairness to unicast sessions [96]. Our scheduling algorithm is based on graph coloring of the contention graph, and it assumes the spatial TDMA MAC protocol [58]. We choose TDMA rather than CSMA/CA because the random nature of CSMA/CA MAC protocols makes QoS support inherently difficult. Spatial TDMA enables collision-free communications and fine control of the throughput and delay of network traffic. Given the contention relationships of the transmissions in the network, our scheduling algorithm produces optimal schedules that achieve the allocated rates.

Our work contains several major contributions: First, our framework provides efficient and fair rate allocation for both unicast and multicast traffic in wireless mesh networks. Second, our framework effectively schedules the allocated rates which results in guaranteed throughput and bounded delay for session recipients. Third, our framework is very efficient and is suitable for a centralized implementation.

4.1 System Model

4.1.1 Routing Tree and Transmission Set

Our framework only deals with the aggregated traffic between the mesh routers and the gateway; we assume that mesh routers employ appropriate methods to distribute traffic to their clients. For simplicity, we only consider downlink traffic from the gateway node to mesh routers. However, our framework can be applied to uplink traffic as well.

In this chapter, we assume that the routing information is given to our framework. Since we only consider traffic from the Internet gateway to mesh routers, we use the Weighted Connected Dominating Set (WCDS) algorithm proposed in [61] to construct a broadcast tree and then prune the tree to accommodate all the multicast and unicast sessions. We denote the resulting tree structure as the *routing tree*. Note that our framework can work with other routing schemes. For example, we can also accommodate using a separate multicast tree for each multicast session as was done in [60].

Figure 4.1 shows an example scenario we use to describe our framework. In this figure, solid lines depict links belonging to the routing tree while dashed lines depict other potential links. We denote the set of network sessions as S . For this example scenario, two network sessions s_0 and s_1 are defined. The source of s_0 is node 0; the recipients of s_0 are nodes 3, 6 and 7. The source of s_1 is node 0; the recipient of s_1 is node 4. We also specify the rate of each link in the routing tree, where symbol b represents the minimum link rate supported by the network.

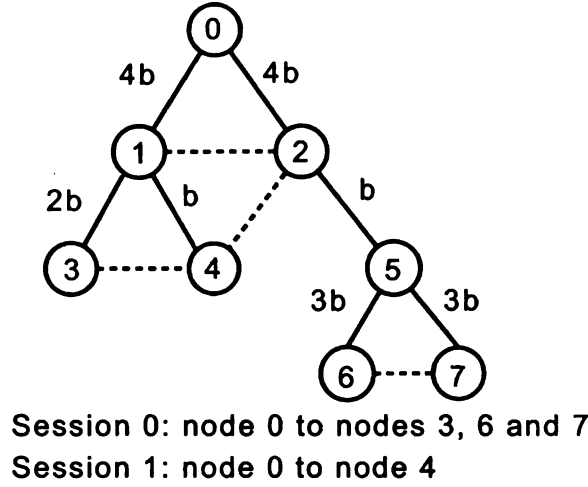


Figure 4.1. An Example Scenario

In order to model multicast sessions, we define the notion of *transmissions*, which is a generalization of *links*. A transmission consists of the following attributes: (1) A sender; (2) A list of recipients; (3) The session carried by this transmission; (4) Transmission rate used. Note that we can also allow a transmission to carry multiple sessions. However, in order to accommodate scenarios where multiple multicast trees are constructed, a transmission is only associated with a single session in this chapter.

MAC protocols such as IEEE 802.11 do not support multi-rate multicast and mandate that all link-layer broadcasts use the same rate. In [60], all link-layer broadcasts proceed with the rate of 2Mbps. However, newer MAC protocols such as IEEE 802.16 support multi-rate multicast at the link layer [61]. In this chapter, we assume a MAC protocol that supports multi-rate multicast by employing different transmission rates for link-layer broadcasting. The rate of a transmission is the minimum rate of all the links between the sender and the recipients [61]. The link rate depends on the modulation and coding scheme used for the link, which is determined by the Signal

to Noise Ratio (SNR) at the receiver side of the link.

We denote the set of transmissions as T . Given the routing tree R and the session set S , our algorithm to construct transmission set T is shown in Algorithm 1. For routing schemes where multiple multicast trees are constructed, we can execute the above algorithm for each multicast tree and aggregate the resulting transmissions. For the example scenario shown in Figure 4.1, the algorithm outputs 6 transmissions as shown in Table 4.1.

Algorithm 1. ConstructTransmissionSet(R, S)

```

1  $T = \emptyset$ 
2 for each session  $s$  in  $S$  do
3   for each node  $n$  in  $R$  that is a recipient of  $s$  do
4      $\quad$  Add session  $s$  to node  $n$ 's session list
5 Use the Breadth First Search (BFS) algorithm to partition the nodes in  $R$  into
   different levels  $l_0, \dots, l_m$ 
6 for  $l = l_m$  to  $l_0$  do
7   for each node  $n$  at level  $l$  do
8      $\quad$  Add all the sessions in the session lists of node  $n$ 's children to node  $n$ 's
        $\quad$  session list
9 for each node  $n$  in  $R$  do
10  if  $n$  has children then
11    for each session  $s$  in the session list of node  $n$  do
12      Construct a new transmission  $t$ 
13      Add each child of  $n$  which has session  $s$  in its session list into the
       recipient list of  $t$ 
14      Add transmission  $t$  into  $T$ 
15 return  $T$ 

```

Table 4.1. An Example Transmission Set

Transmission ID	0	1	2	3	4	5
Sender	0	0	1	1	2	5
Recipient List	1, 2	1	3	4	5	6,7
Session List	0	1	0	1	0	0
Transmission Rate	4b	4b	2b	b	b	3b

4.1.2 Transmission Contention Graph

In [97], Jain, et al. defined two contention models: Protocol Model and Physical Model. In [61], Chou, et al. proposed and justified a contention model for multi-rate multicast based on the Protocol Model. Under this model, every link-layer transmission rate corresponds to a fixed transmission range and interference range.

In addition, transmission i contends with transmission j if (1) Both transmissions share the same sender; (2) Any recipient of transmission i is the sender of transmission j or vice versa; (3) Any recipient of transmission i is within the interference range of the sender of the transmission j or vice versa. The first two conditions are referred to as *primary interferences*. The last condition is referred to as *secondary interference*. The secondary interference can be eliminated by using orthogonal channels.

According to the above contention model, we can define a *transmission contention graph* $G_c(V_c, E_c)$ based on the contention relationship among different transmissions. Contention graphs have been used in several papers [46, 97] to model network resource constraints. The vertex set V_c contains all the transmissions in the network. An edge in set E_c indicates that two vertices (transmissions) contend with each other. Figure 4.2 shows the contention graph of our example scenario considering both primary interference and secondary interference. Figure 4.3 shows the contention graph of our

example scenario considering only primary interference. Nodes in the figure represent the transmissions defined in Table 4.1. We specify the sender and the receivers in each node.

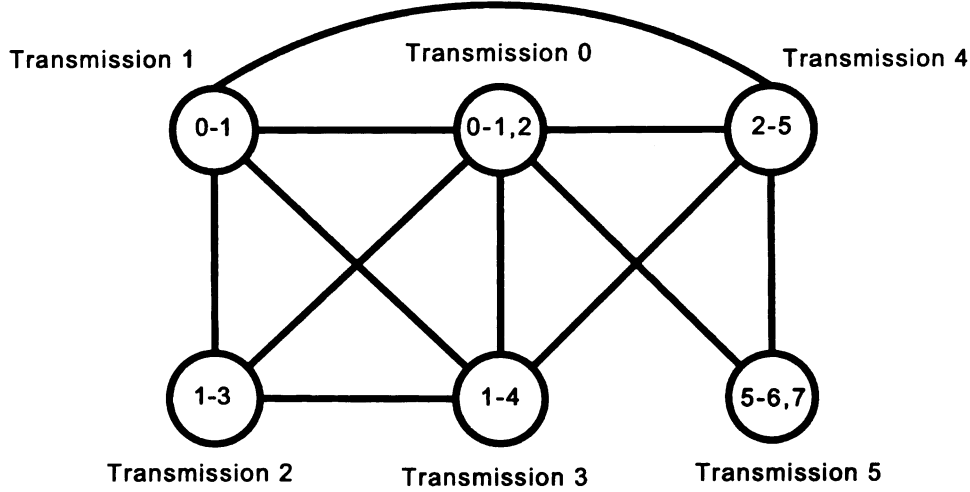


Figure 4.2. An Example Contention Graph with both Primary and Secondary Interference

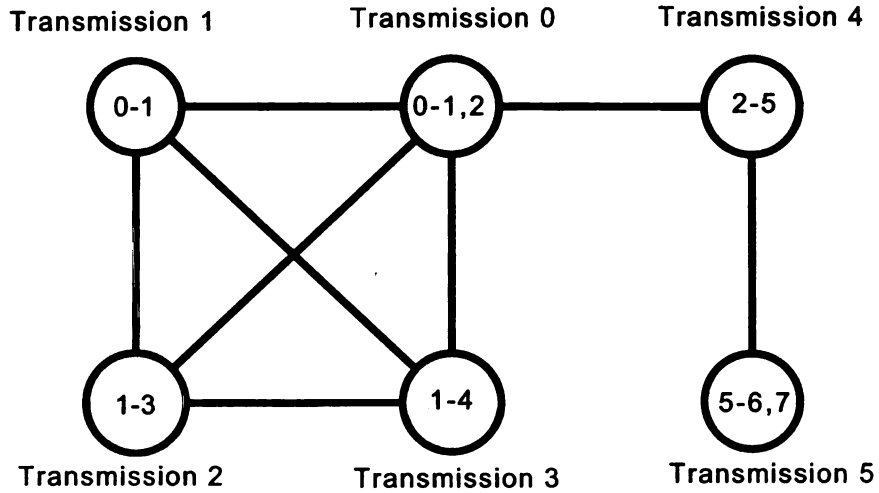


Figure 4.3. An Example Contention Graph with only Primary Interference

We are interested in *maximal cliques* and *maximal independent sets* of the contention graph. A set of nodes is a clique if its induced subgraph is complete. A maximal clique is defined as a clique that is not contained in any other clique. We

denote the set of maximal cliques of a contention graph as C . Also, a *maximum clique* is defined as the largest clique of the graph. In the transmission contention graph given in Figure 4.2, we have three maximal cliques, $C_1 = \{0, 1, 2, 3\}$, $C_2 = \{0, 1, 3, 4\}$ and $C_3 = \{0, 4, 5\}$. An independent set is a set of nodes that are not connected by any edges, and a maximal independent set is an independent set not contained by any other independent set.

4.1.3 Clique Feasibility and Scheduling Feasibility

For contention graph G_c , let I be the family of all independent sets of this graph. Schedule D can be defined as an infinite sequence of independent sets, $I_1, I_2, \dots, I_k, \dots$, where $I_k \in I$ [98, 47]. The frequency of transmission i in schedule D is then defined as:

$$f_i = \lim_{t \rightarrow \infty} \frac{\sum_{k=1}^{k=t} D(i, k)}{t}$$

where $D(i, k)$ is an indicator function such that, $D(i, k) = 1$ if $i \in I_k$, and $D(i, k) = 0$ otherwise. We say that a schedule D is periodic with period N , if $I_j = I_{N+j} = I_{2N+j} = \dots$ for all $1 \leq j \leq N$ [98].

For a transmission contention graph with M transmissions, a vector of frequencies $\mathbf{f} = (f_1, \dots, f_M)$ is feasible if there exists a schedule D such that the frequency of the i th transmission in schedule D is at least f_i for $1 \leq i \leq M$ [98]. The frequency of transmission i can be treated as the normalized bandwidth allocated to this transmission in schedule D [47]. A vector of frequencies \mathbf{f} is *clique feasible* [98] for the

contention graph G_c if

$$\sum_{i \in C_j} f_i \leq 1, \forall C_j \in C$$

It has been shown that if the contention graph is a perfect graph, clique feasibility is equivalent to *scheduling feasibility* of the frequency vector [98, 99].

In graph theory, a *perfect graph* is a graph in which every induced subgraph can be colored with p colors where p is the size of the maximum clique in the subgraph. A *chordal graph*, a graph that does not contain an induced k -cycle for $k \geq 4$, is a perfect graph. Chordal graphs are sometimes called *triangulated graphs*. An alternative definition for chordal graph is related to simplicial elimination ordering. A vertex v of a graph G is called *simplicial* if its neighbors in G form a clique. A *simplicial elimination ordering* of G is a vertex order $v_1, \dots, v_i, \dots, v_n$ such that every vertex v_i is a simplicial vertex in the subgraph induced by $\{v_1, \dots, v_i\}$ [100]. A graph is a chordal graph if and only if it has a simplicial elimination ordering [100].

We can use the lexicographic BFS algorithm [101] for recognizing a chordal graph and outputting a simplicial elimination ordering with time complexity $O(m + n)$, where m is the number of edges and the n is the number of vertices of the input graph. A general contention graph is not always chordal graph. However, in [102], the authors proved that for a tree-structured wireless mesh network with only primary interference, its contention graph is always a chordal graph. Although the authors consider network scenarios with only unicast sessions, we can apply the result to our work as well. First, we construct a link contention graph assuming that each transmission consists of one or more links. The contention graph is shown in Figure

4.4. According to the result in [102], this graph is always a chordal graph. This contention graph can be transformed to the transmission contention graph as shown in Figure 4.3 by grouping nodes (links) that belong to the same transmission. The grouping process guarantees that the resulting transmission contention graph is still a chordal graph.

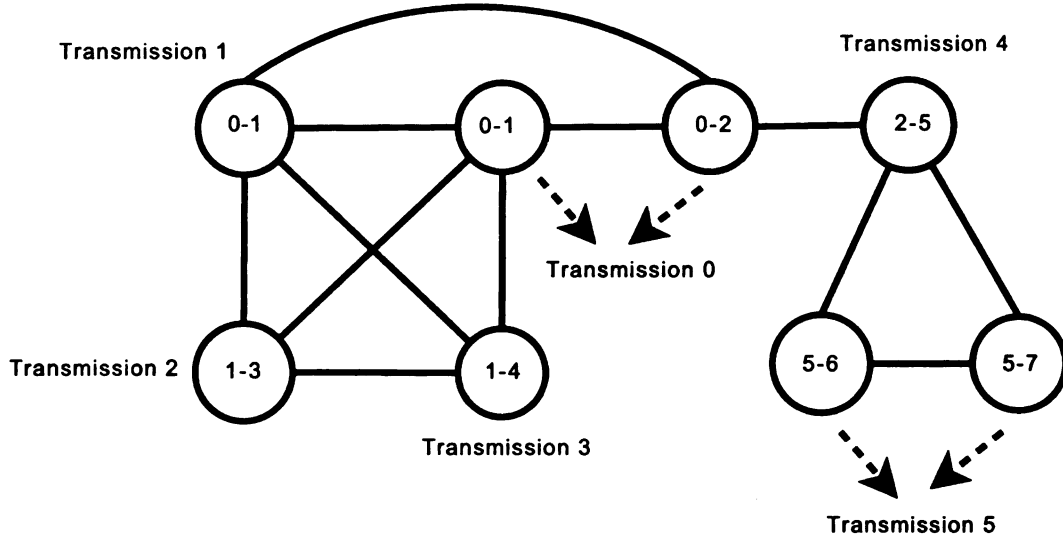


Figure 4.4. An Example Link Contention Graph

In the case where secondary interference cannot be eliminated, we can transform a general graph to a chordal graph in $O(mn)$ time using the LEX M minimal triangulation algorithm [103]. Recently, an $O(n^{2.69})$ implementation of LEX M has been proposed in [104]. In this chapter, in order to maintain the scheduling feasibility of the frequency vector, we apply the LEX M algorithm to transform a general contention graph to a chordal graph. Since the transformation only adds edges to the contention graph, the resulting contention graph will not introduce any violation of existing contention relationships. In fact it adds unnecessary contention relationships, which results in more conservative rate allocation and less spatial reuse. However the

benefits of maintaining chordal graph lie in several aspects: First, a perfect graph enables our algorithm to produce rate allocations that are schedulable. Second, a chordal graph facilitates the optimal scheduling algorithm, which will be introduced in Section 4.2.3. Third, a perfect graph enables $O(m + n)$ time algorithm for obtaining the maximal clique set [101]. For general graphs, algorithms for obtaining the maximal clique set has exponential time complexity [88].

To formally define clique feasibility, we first define matrix $\mathbf{Q} = \{Q_{nm}\}$:

$$Q_{nm} = \begin{cases} 1, & \text{if clique } n \text{ contains transmission } m \\ 0, & \text{otherwise} \end{cases}$$

For the example scenario, with maximal clique set $C = \{C_1, C_2, C_3\}$ and transmission set $T = \{0, 1, 2, 3, 4, 5\}$, we have

$$\mathbf{Q} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

Thus we can express the clique feasibility of a contention graph G_c as follows:

$$\sum_{m: Q_{nm}=1} f_m \leq 1, \quad \forall n \in C$$

In fact, the constraint states that the sum of the frequencies of all the transmissions in a maximal clique, cannot exceed the *normalized capacity* of the clique, which is 1 [47].

In the following, we express the clique feasibility in terms of the session rate vector.

We assume that a fixed-length scheduling period consists of several time slots. Each

time slot has a fixed duration. We use $\mathbf{x} = (x_s, s \in S)$ to denote the *rate allocation vector* of network sessions where x_s is expressed in bits per scheduling period. Assume that transmission m carries session s and the transmission rate is b_m , expressed in bits per time slot. Also denote the total number of time slots in a scheduling period as N . Then we can express the transmission frequency f_m needed to support session rate x_s as follows:

$$f_m = \frac{\lceil x_s/b_m \rceil}{N}$$

In fact, $\lceil x_s/b_m \rceil$ is the number of time slots needed for transmission m to support the session rate x_s . In order to simplify the rate allocation algorithm, we ignore the ceiling function and express f_m as

$$f_m = \frac{x_s/b_m}{N}$$

As a result, the allocated rate x_s may need to be scaled down to x'_s so that $\frac{\lceil x'_s/b_m \rceil}{N} \leq \frac{x_s/b_m}{N}$.

We define a matrix $\mathbf{R} = \{R_{ms}\}$:

$$R_{ms} = \begin{cases} \frac{1}{b_m * N}, & \text{if transmission } m \text{ carries session } s \\ 0, & \text{otherwise} \end{cases}$$

By using the matrix $\mathbf{P} = \mathbf{Q}\mathbf{R}$ ($P_{ns} = Q_{nm}R_{ms}$), we can write the scheduling feasibility in the following form:

$$\mathbf{P}\mathbf{x} < \mathbf{1} \tag{4.1}$$

4.2 Rate Allocation and Scheduling Framework

In [60], the authors proposed a scheduling algorithm named Least Overlapped First (LOF) and a rate allocation strategy for multiple multicast sessions. Thereafter, we use LOF to denote the proposed framework including both the rate allocation and the scheduling algorithms. We choose LOF for comparison since, to the best of our knowledge, it is the only framework that considers both rate allocation and scheduling for multicast traffic in TDMA based wireless mesh networks. Note that LOF does not take multi-rate multicast into consideration. For the purpose of comparison, we extend the LOF framework in Section 4.2.1 to accommodate multi-rate multicast.

In contrast to the LOF framework, our framework first derives the rate allocation using the Network Utility Maximization where the allocation constraint is introduced in Section 4.1.3. We then propose a scheduling algorithm based on Optimal Graph Coloring (OGC). Thereafter, we use OGC to denote our proposed framework including both the rate allocation and the scheduling algorithms.

4.2.1 Least Overlapped First

LOF first enumerates all the independent sets in the contention graph. Note that LOF works with the original contention graph without triangulation. For a successful schedule, all transmissions need to be included and each transmission can be included only once. To maximize spatial reuse, LOF tries to construct a schedule using as few independent sets as possible.

Each independent set is assigned a rank, which is equal to the number of common

transmissions this set has with all other independent sets of the same size. At each step of the scheduling algorithm, LOF selects the independent set which has the maximum number of transmissions that do not intersect with the members of the existing schedule. If there is more than one independent set with the same number of non-intersecting transmissions, LOF selects the one with the lowest rank to be added to the schedule breaking ties arbitrarily. LOF terminates when all transmissions are included in the schedule.

In [60], all sessions are implicitly assigned the same rate, which is determined by the length of the scheduling period and the duration of a time slot. We generalize LOF to make it work with different link rates and a fixed length scheduling period. For every independent set added into the schedule, define its rate as the lowest rate among all the transmissions in the set. Denote a schedule as $I = \{I_1, \dots, I_K\}$, where I_i represents the independent sets that are included in the schedule. For each independent set I_i in the schedule, denote its rate as b_i . The session rate r is then determined by solving the following equation

$$\sum_{i=1}^K \frac{r}{b_i} = N$$

where N is the total number of available time slots in a scheduling period.

For the example scenario in Figure 4.1 and its corresponding contention graph in Figure 4.2, the set of all independent sets is: $\{\{0\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{2,4\}, \{2,5\}, \{3,5\}, \{1,5\}\}$. Independent set $\{2,4\}$ has rank 1; independent sets $\{3,5\}$ and $\{1,5\}$ have rank 2 and independent set $\{2,5\}$ has rank 3. LOF will construct the schedule $\{\{2,4\}, \{3,5\}, \{0\}, \{1\}\}$. Assuming that there are 100 time slots in a

scheduling period, the session rate vector is $(40b, 40b)$. The schedule of a period is shown in Figure 4.5.

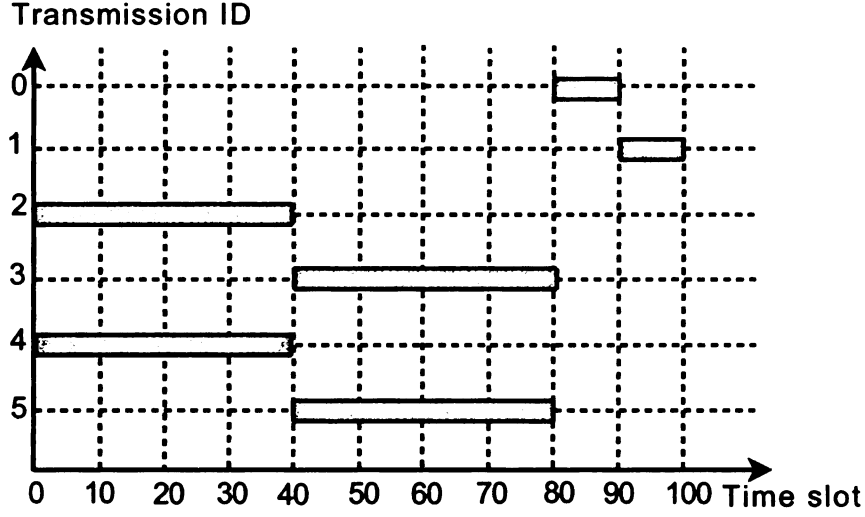


Figure 4.5. The Schedule Obtained by LOF

4.2.2 Utility Maximization Based Rate Allocation

We develop our rate allocation algorithm based on Network Utility Maximization (NUM) [52]. We associate a utility function $U_s(x_s)$ to every session $s \in S$. Similar to [96], we cast the problem of rate allocation as that of maximizing the total utility of the network. When we consider both unicast and multicast sessions, it has been pointed out that it is appropriate to use *receiver-oriented* utility functions such as $U_s(x_s) = r_s u(x; \alpha)$, where r_s is the number of recipient of session s and $u(x; \alpha)$ is defined as follows [96]:

$$u(x; \alpha) = \begin{cases} \ln(x), & \alpha = 0 \\ -\frac{x^{-\alpha}}{\alpha}, & \alpha > 0 \end{cases}$$

The parameter α provides a mechanism to tune the unfairness to unicast while retaining the incentive to use multicast [96]. In this chapter, we choose $\alpha = 0$ for the following reasons: First, it provides incentive to use multicast since some amount of unfairness to unicast is achieved [96]. Second, when our framework is applied to networks containing only unicast traffic, it achieves proportional fairness among unicast traffic [105]. However, our framework can also accommodate other values of α if the resulting utility function is concave.

The utility maximization objective can be formally written as

$$\mathbf{P}: \max_{x_s \geq 0} \sum_{s \in S} U_s(x_s), \text{ subject to } \mathbf{P}\mathbf{x} \leq 1 \quad (4.2)$$

The above problem is referred to as the system primal problem. It is a typical convex optimization problem [84], which can be solved by using the Lagrange duality [44, 45]. The dual problem is defined as

$$\mathbf{D}: \min_{\lambda \geq 0} D(\lambda) \quad (4.3)$$

where

$$\begin{aligned} D(\lambda) &= \max_{x_s \geq 0} \left(\sum_{s \in S} U_s(x_s) - \lambda^T (\mathbf{P}\mathbf{x} - 1) \right) \\ &= \sum_{s \in S} \max_{x_s \geq 0} (U_s(x_s) - x_s \sum_{n \in C} \lambda_n P_{ns}) + \sum_{n \in C} \lambda_n \end{aligned}$$

Let us define

$$\mu_s = \sum_{n \in C} \lambda_n P_{ns}$$

In fact, the Lagrange multiplier λ_n can be interpreted as the shadow price [44] of clique n . Also $x_s \mu_s$ may be interpreted as the total price charged for session s for transmitting at rate x_s .

To solve the dual problem, we use an iterative algorithm with the help of the gradient projection method [49]. The algorithm is described in Algorithm 2. A similar algorithm is used in [49]. The algorithm takes the inputs of the maximal clique set C , session set S , initial vector \mathbf{x} and $\boldsymbol{\lambda}$.

Algorithm 2. RateAllocation($C, S, \mathbf{x}, \boldsymbol{\lambda}$)

```

1  $t = 0$  /*  $t$  is the iteration number */
2  $\mathbf{x}(t) = \mathbf{x}; \boldsymbol{\lambda}(t) = \boldsymbol{\lambda}$ 
3 while  $\mathbf{x}$  does not converge do
4   for each maximal clique  $n \in C$  do
5     Update  $\lambda_n$  as  $\lambda_n(t+1) = [\lambda_n(t) - \gamma \frac{\partial D(\boldsymbol{\lambda}(t))}{\partial \lambda_n}]^+$  /*  $\gamma$  is the step
       size */
6     In our case,  $\lambda_n(t+1) = [\lambda_n(t) - \gamma(1 - \sum_{s \in S} x_s(t) P_{ns})]^+$ 
7   for each session  $s \in S$  do
8     Update  $x_s$  by solving the problem  $\max_{x_s \geq 0} (U_f(x_s) - x_s \mu_s)$ 
9     In our case,  $x_s(t+1) = \frac{r_s}{\mu_s}$ 
10   $t = t + 1$ 

```

By choosing an appropriate step size γ , starting from any initial vector \mathbf{x} and $\boldsymbol{\lambda}$, the above iterative algorithm will converge to the optimal solution $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ [46, 49]. Furthermore, the solution is primal-dual optimal, which means \mathbf{x}^* is also the optimal rate vector for the primal problem. The optimal solution can be obtained in worst-case polynomial-time complexity [84, 90].

4.2.3 Optimal Graph Coloring Based Scheduling

Vertex coloring of a graph assigns different colors to vertices so that no two adjacent vertices have the same color. The greedy coloring algorithm [100] is a heuristic for graph coloring with $O(m+n)$ time complexity. If we give greedy coloring a simplicial

elimination ordering of the vertices, then the greedy algorithm yields an optimal coloring [106]. In other words, greedy graph coloring is optimal for chordal graphs [106].

Algorithm 3. OptimalGraphColoring(G_c, T, P)

```

1  $G_e = \emptyset$ 
2 for each vertex  $v$  in  $G_c$  do
    // Denote  $t_v$  as the ID the transmission that  $v$  corresponds to
    // Denote  $n_v$  as the number of time slots needed for
    // transmission  $t_v$ 
3    $n_v = \lceil \frac{x_s/bm}{N} \rceil$ , where  $x_s$  is the rate of the session carried by transmission  $t_v$ 
4   Add  $n_v$  vertices to  $G_e$ , and their corresponding transmission is  $t_v$ 
    // Denote these group of vertices as  $O_v$ 
5   Add edges between any pair of these vertices
6 for each edge  $e : (v_i, v_j)$  in  $G_c$  do
7   Add edges between nodes in  $O_{v_i}$  and  $O_{v_j}$ 
    //  $G_e$  is still a chordal graph based on its construction
8 Get a simplicial elimination ordering  $\nu$  of  $G_e$  using the lexicographic BFS
   algorithm
   // The following in the greedy graph coloring algorithm
9 for  $i = 0$  to  $|\nu|$  do
10   $Q(\nu(i)) =$  nodes that are neighbors of  $\nu(i)$ 
11   $p =$  lowest time slot in  $P$  that is not used in  $Q(\nu(i))$ 
12  Assign time slot  $p$  to vertex  $\nu(i)$ 
13  Set transmission  $t_{\nu(i)}$  to be active at time slot  $p$ 

```

Based on the above discussion, we propose a scheduling algorithm based on a greedy vertex coloring of the contention graph. The algorithm is described in Algorithm 3 where time slots are considered as colors to be assigned to different transmissions. Denote the set of time slots in a scheduling period as $P = \{1, \dots, N\}$. The algorithm takes the inputs of the triangulated contention graph G_c , the transmission set T and the set of time slots P .

If the scheduling period contains N time slots, we can prove that graph G_e can be colored using at most N colors, which means that all transmissions can be scheduled in at most N time slots. In this case, we say that the scheduling algorithm realizes the rate allocation. The proof is given as follows:

According to the definition of perfect graph given by Claude Berge, perfect graph G can be colored with $\omega(G)$ colors, where $\omega(G)$ is the size of the largest clique of G [99]. According to the construction of G_e , every maximal clique of G_e corresponds to a maximal clique of G_c . Assume that the largest clique of G_e is constructed from maximal clique C_j of G_c . The size of the largest clique of G_e can be calculated as

$$\sum_{i \in C_j} S_i$$

where S_i is the number of time slots allocated to the transmission that corresponds to vertex i in G_c . Our rate allocation satisfies the clique feasibility constraint defined in Section 4.1.3:

$$\sum_{i \in C_j} f_i \leq 1, \forall C_j \in C$$

where C is the set of maximal cliques of graph G_c . In addition, Section 4.1.3 shows that $S_i \leq f_i * N$. So we have

$$\sum_{i \in C_j} S_i \leq \sum_{i \in C_j} f_i * N \leq N$$

Thus we have proved that the size of the largest clique of G_e is equal to or smaller than N . Therefore, all transmissions can be scheduled in at most N time slots.

Assume that a scheduling period contains 100 time slots. For the example scenario in Figure 4.1 and its corresponding contention graph in Figure 4.2, our rate allocation

algorithm yields the session rate vector $(60b, 20b)$. The schedule obtained by OGC is shown in Figure 4.6.

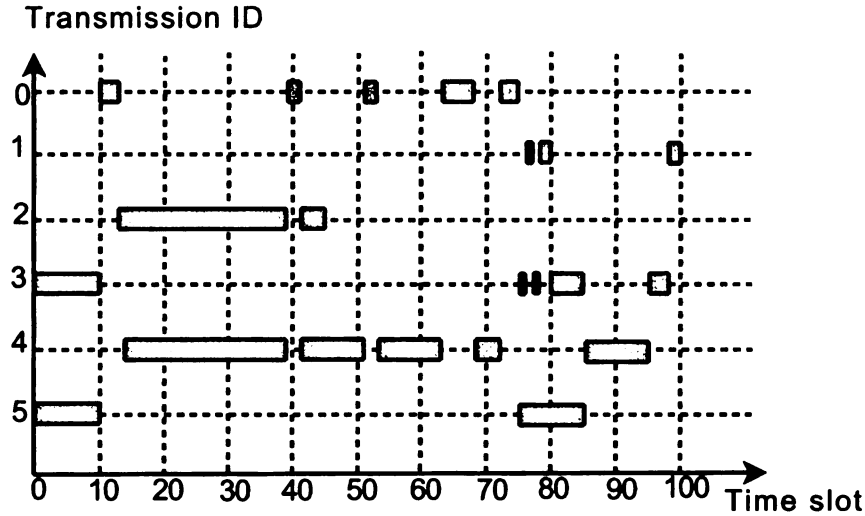


Figure 4.6. The Schedule Obtained by OGC

4.3 Framework Extensions

4.3.1 Real-time Session Rate Allocation

As discussed in Chapter 3, we can classify network sessions as real-time and elastic. In this section, we propose an admission control and bandwidth reservation algorithm for the real-time sessions similar to the algorithm proposed in Chapter 3.

According to the session classification, the network session set S is divided into real-time session set S^r and elastic session set S^e . We perform admission control and bandwidth reservation for real-time sessions first and then allocate the left-over bandwidth to elastic sessions using our rate allocation algorithm introduced in Section 4.2.2. Real-time sessions are processed sequentially. We assume that each real-time

session $s \in S^r$ has a rate requirement of r_s . It is admitted to the network if

$$P_{ns}r_s < 1, \quad \forall n \in C$$

which means that the requested rate can be supported by all the maximal cliques. When admitted, the allocated rate x_s equals to the requested rate r_s . Also, the normalized capacity of every clique n is updated to $1 - P_{ns}x_s$, which means capacity has been reserved for session s . When we perform the rate allocation for elastic flows, the rates allocated for real-time session are fixed.

The admission control algorithm can be more flexible. For example, in order not to starve elastic sessions, a certain proportion of the normalized capacity of each clique can be reserved for elastic sessions before the admission control process.

4.3.2 Channel Assignment

In Section 4.1.3 we have shown that if secondary interference can be eliminated using multiple orthogonal channels, the resulting contention graph is chordal. Thus no triangulation of the original contention graph is needed for performing our rate allocation and scheduling algorithms. In this section we propose a channel assignment algorithm to eliminate the secondary interference in the network.

The output of the scheduling algorithm indicates which group of transmissions will be active simultaneously at each time slot. For each time slot, we need to assign channels to this group of transmissions in order to eliminate secondary interference. Denote the set of channels available in the network as Q . Our channel assignment algorithm is described in Algorithm 4. The input of the algorithm is the transmission

set T and the channel set Q . Note that only secondary interference is possible among transmissions that are active in the same time slot.

Algorithm 4: ChannelAssignment(T, Q)

```

1 for  $t = 1$  to  $|T|$  do
2    $Q(t) =$  transmissions in  $T$  that interfere with  $t$ 
3    $p =$  lowest channel in  $P$  that is not assigned in  $Q(t)$ 
4   Assign channel  $p$  to transmission  $t$ 

```

4.4 Performance Evaluation

We use the ns-2 [107] simulator for performance evaluation. In our simulation, we use a TDMA MAC protocol similar to IEEE 802.16. A MAC frame consists of a control subframe and a data subframe, as shown in Figure 4.7. The duration of a frame corresponds to the scheduling period defined in our scheduling algorithm. A frame is divided into fixed-length time slots.

Similar to the centralized scheduling scheme used in IEEE 802.16 mesh networks, we implement our framework in a centralized manner. The benefits of a centralized implementation of the NUM based rate allocation has been discussed in Chapter 3.

In a control subframe, via control message exchange, mesh routers join multicast sessions or establish a unicast session between the gateway node and itself. The gateway node then constructs the routing tree, performs rate allocation and establishes the network wide schedule. The schedule is propagated to mesh routers during the control subframe. Our rate allocation algorithm (including transmission set and perfect contention graph construction) is a polynomial-time algorithm while our scheduling

algorithm is a linear time algorithm. In addition our framework operates on aggregated traffic between the mesh routers and the gateway. Thus our framework is very efficient and is suitable for a centralized implementation. Since our framework can be executed at the beginning of every frame, it can accommodate dynamic scenarios where network traffic changes frequently.

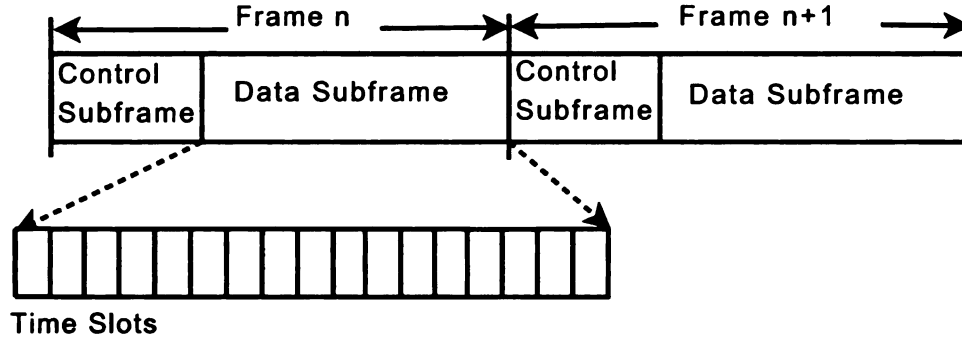


Figure 4.7. MAC Frame Structure

The detailed simulation parameters related to the physical and MAC layer properties are given in Table 4.2. Table 4.3 shows how the type of modulation and link rate are related to the received SNR [62]. CBR is used as the traffic source. We use aggregated throughput, average delay and total utility as the evaluation metrics. We compare the performance of LOF and OGC under different network scenarios.

4.4.1 Real-time Traffic Rate Allocation Evaluation

We first evaluate our framework in scenarios where real-time sessions exist. We simulate a flat area of 1000m by 1000m with 30 randomly positioned stationary mesh routers. A gateway node is placed in the center of the topology. Figure 4.8 shows the example scenario. Node 0 is the gateway node. There are five network sessions in

Table 4.2. Simulation Parameters

Parameter	Value
Frequency band (GHz)	5
Channel bandwidth (MHz)	20
Frame duration (ms)	20
Time slot duration (μ s)	50
Transmit power (dBm)	20
Path Loss Model	$28.3 \log(d) + 41.9$, d is the distance
Noise Level (dBm)	$\text{bandwidth} * \frac{4 * 10^{-12}}{10^9}$, according to [108]

Table 4.3. Modulation Type and Link Rate vs. SNR

Modulation	Bit-rate (Mbps)	Received SNR (dB)
BPSK 1/2	7.68	3
QPSK 1/2	15.36	6
QPSK 3/4	23.04	8.5
16QAM 1/2	30.72	11.5
16QAM 3/4	46.08	15
64QAM 2/3	61.44	19
64QAM 3/4	69.12	21

the network, which is listed in Table 4.4. Session 0 and 1 are real-time sessions with specific rate requirements.

In our simulation, we vary the rate requirements of real-time sessions and measure the throughput of each real-time session receiver and the aggregate throughput of the receivers of elastic sessions. The simulation result is shown in Figure 4.9. The rate requirements of real-time sessions are always satisfied: the throughput of each receiver equals to the rate requirement of its associated session. With the increase of the rate requirement, the aggregate throughput of elastic sessions get reduced accordingly.

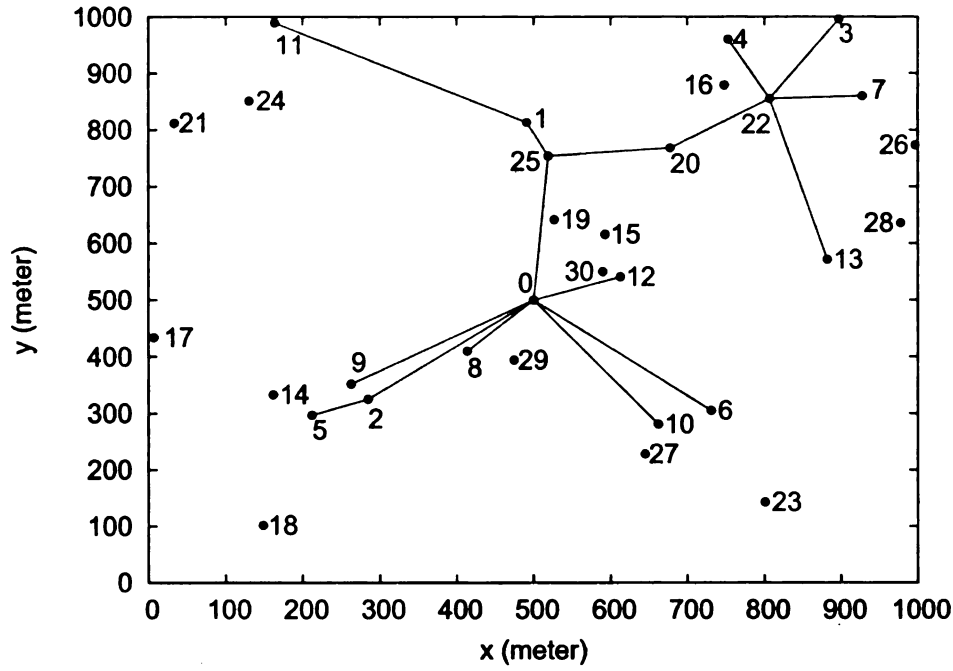


Figure 4.8. Simulation Scenario

Table 4.4. Session List

Session Number	Receivers
0	1, 2, 3, 4
1	5
2	6, 7, 8, 9
3	10, 11
4	12, 13

4.4.2 Simulation Results for Single-channel Scenarios

The simulation topology is similar to the one used in Section 4.4.1. However, our results in this section represents the average performance over 100 randomly generated topologies.

In the first set of simulations, we vary the number of sessions in the networks from 2 to 6. In our simulations, a frame consists of 400 time slots where 300 of them are allocated to downlink data traffic. The source of these sessions is the gateway node,

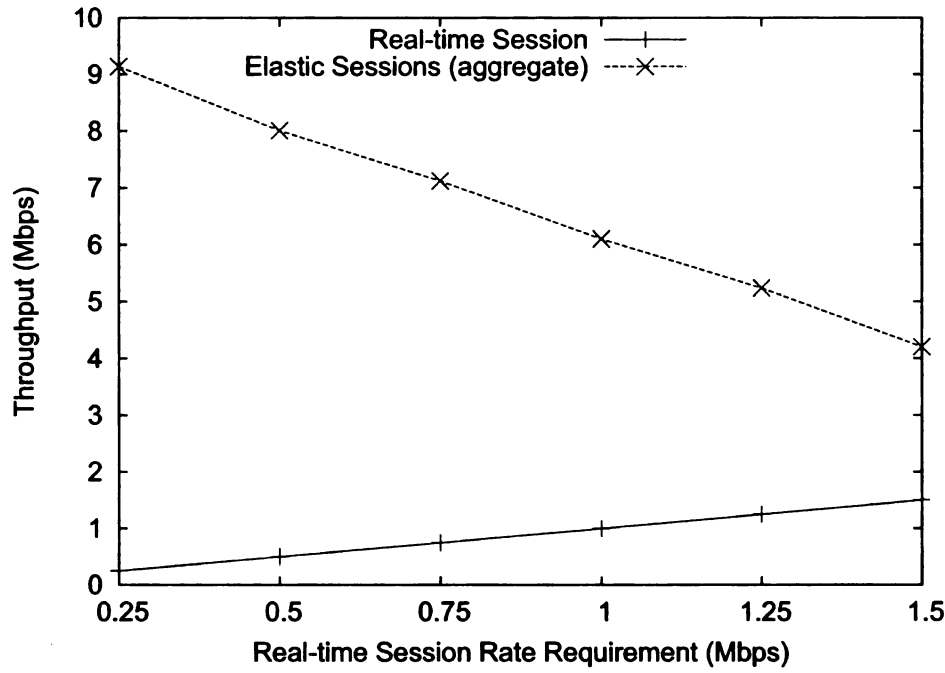


Figure 4.9. Throughput vs. Traffic Request

and we randomly choose 1 to 5 nodes as the recipients for each session. The aggregated throughput of session recipients achieved by using LOF and OGC under different number of sessions are compared in Figure 4.10. The performance improvement of OGC over LOF varies from 25% to 30%. LOF fails to take advantage of multi-rate multicast and guarantees equal throughput to all receivers. OGC achieves higher aggregated throughput by maximizing the total utility of the network and using a link rate aware scheduling algorithm. In our simulations, we observe that for both LOF and OGC, the throughput of a recipient is always equal to the allocated rate of its corresponding session. This demonstrates that both LOF and OGC effectively schedule the allocated rates.

Figure 4.11 compares the average delay of recipients achieved by using LOF and OGC. OGC results in a much smaller average delay than LOF. Neither LOF nor

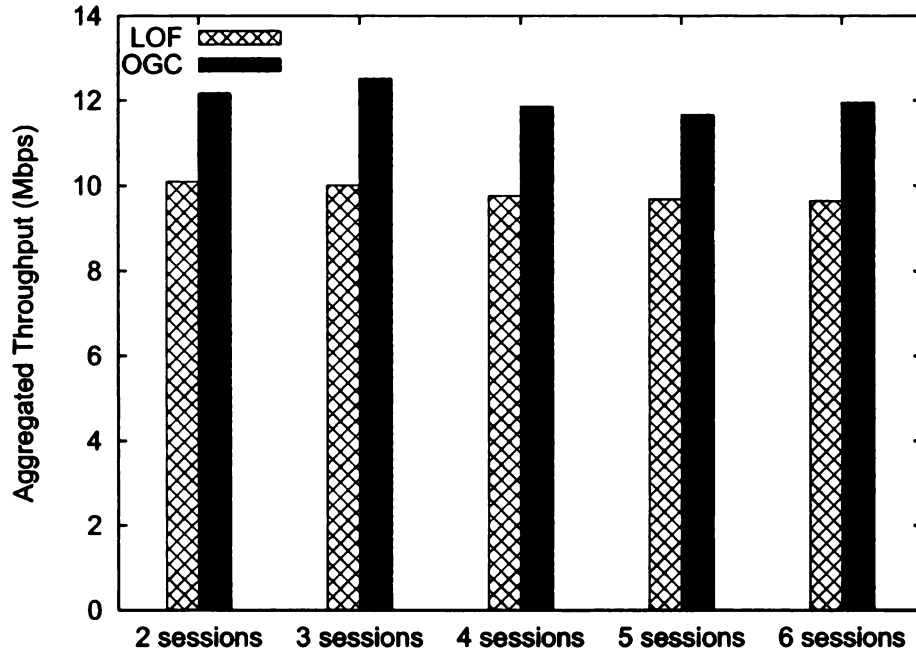


Figure 4.10. Aggregated Throughput vs. Number of Sessions

OGC considers the ordering of transmissions in a frame to reduce the chances that mesh routers transmit before receiving data from the parent node in the routing tree. However, since OGC schedules a transmission multiple times in a frame while LOF schedules a transmission only once in a frame, transmission order has much less impact on OGC than on LOF. Figure 4.12 shows the total utility achieved by using LOF and OGC in different scenarios. OGC consistently achieves higher utility than LOF since the rate allocation in OGC is based on maximizing total utility.

We conduct another set of simulations by varying the number of time slots allocated for downlink data traffic from 150 to 350 time slots. In this set of simulations, we fix the number of sessions in the network to be 4. The aggregated throughput of session recipients achieved by using LOF and OGC under different numbers of time slots are compared in Figure 4.13. The performance improvement of OGC over

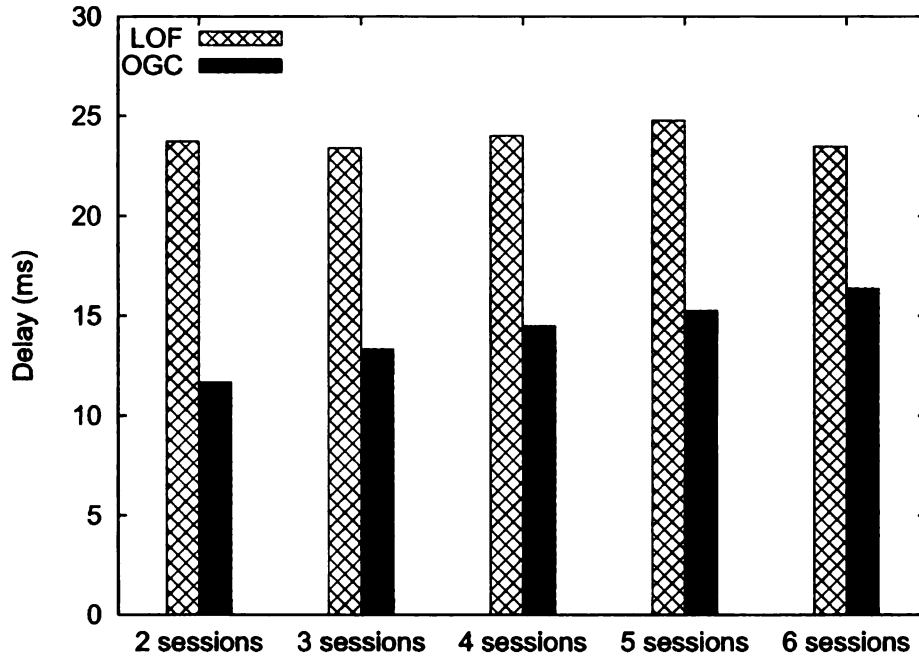


Figure 4.11. Average Delay vs. Number of Sessions

LOF varies from 20% to 25%. In addition, aggregated throughput increases with the number of available time slots. Similar to the previous set of simulations, for both LOF and OGC, the throughput of a recipient is always equal to the allocated rate of its corresponding session.

Figures 4.14 and 4.15 show the average delay and total utility achieved by using LOF and OGC under different numbers of time slots. Similar to the previous set of simulations, OGC consistently achieves lower delay and higher utility than LOF.

4.4.3 Simulation Results for Multi-channel Scenarios

We conduct two sets of simulations similar to the previous subsection. The aggregated throughput of session recipients achieved by using LOF and OGC under different number of sessions are compared in Figure 4.16. The performance improvement of

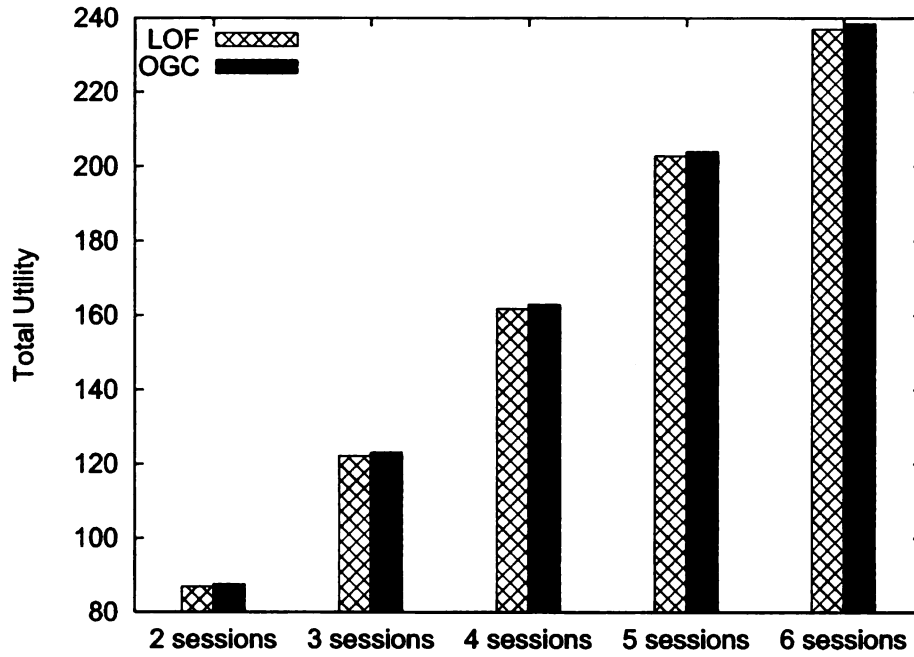


Figure 4.12. Total Utility vs. Number of Sessions

OGC over LOF varies from 30% to 50%. Figure 4.17 compares the average delay of recipients achieved by using LOF and OGC. Figure 4.18 shows the total utility achieved by using LOF and OGC in different scenarios.

The aggregated throughput of session recipients achieved by using LOF and OGC under different numbers of time slots are compared in Figure 4.19. The performance improvement of OGC over LOF is around 50%. Figures 4.20 and 4.21 show the average delay and total utility achieved by using LOF and OGC under different numbers of time slots.

Compared with simulation results for single-channel scenarios, the performance improvement of OGC over LOF is higher. This is because that the original contention is always chordal, with all secondary interference eliminated by multiple orthogonal channels.

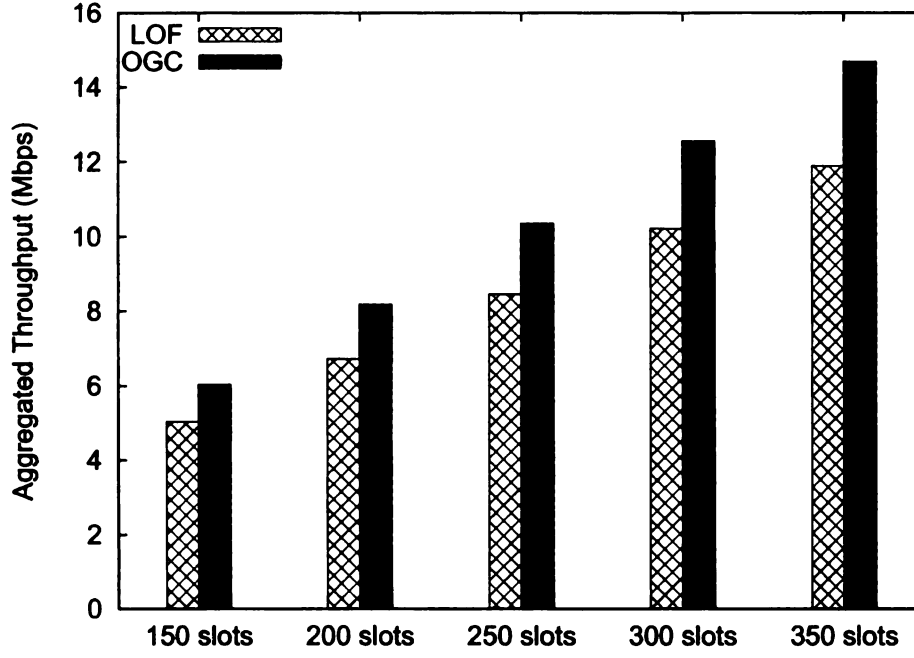


Figure 4.13. Aggregated Throughput vs. Number of Time Slots

4.5 Summary

In this chapter we proposed a framework for rate allocation and scheduling in TDMA based wireless mesh networks. By using transmission contention graphs to model the resource allocation constraints and by maximizing the total utility of the network, our framework provides fair and efficient rate allocation to multicast and unicast sessions. The proposed graph coloring based scheduling algorithm produces an optimal schedule to support the allocated rates. By maintaining perfect contention graphs, our framework enables rate allocation and scheduling with low time complexity. In addition, we discussed admission control of real-time traffic in the network and proposed channel assignment algorithm to eliminate secondary interference when multiple orthogonal channels are available in the network. Our framework significantly outperforms LOF in terms of throughput and delay in our simulation results.

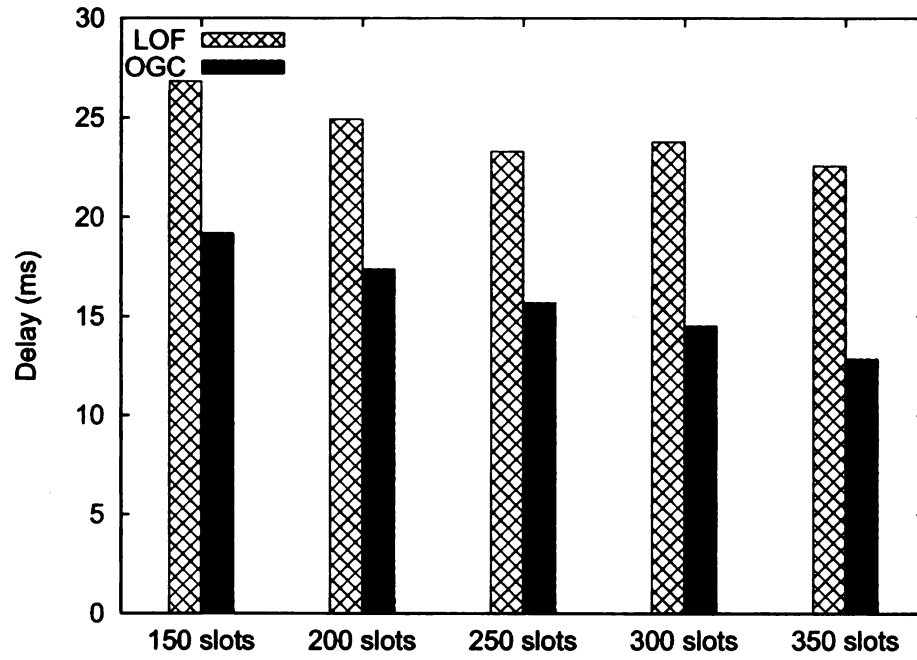


Figure 4.14. Average Delay vs. Number of Time Slots

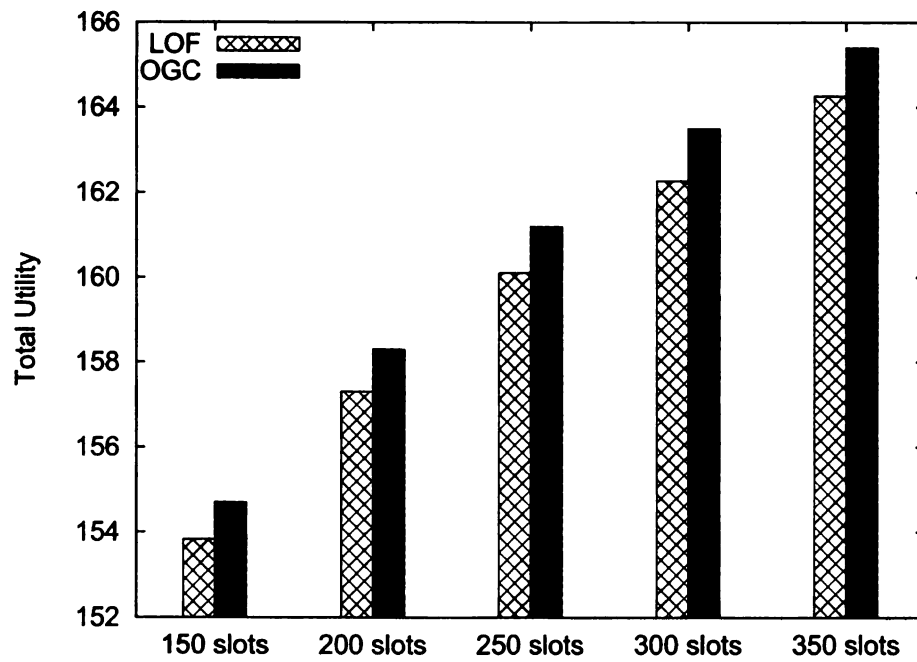


Figure 4.15. Total Utility vs. Number of Time Slots

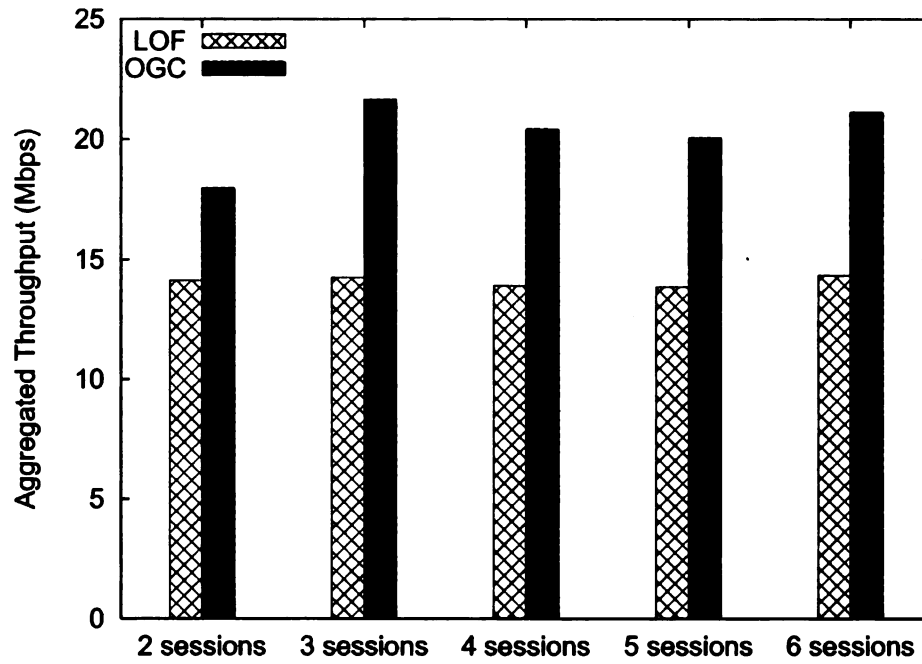


Figure 4.16. Aggregated Throughput vs. Number of Sessions (Multiple Channels)

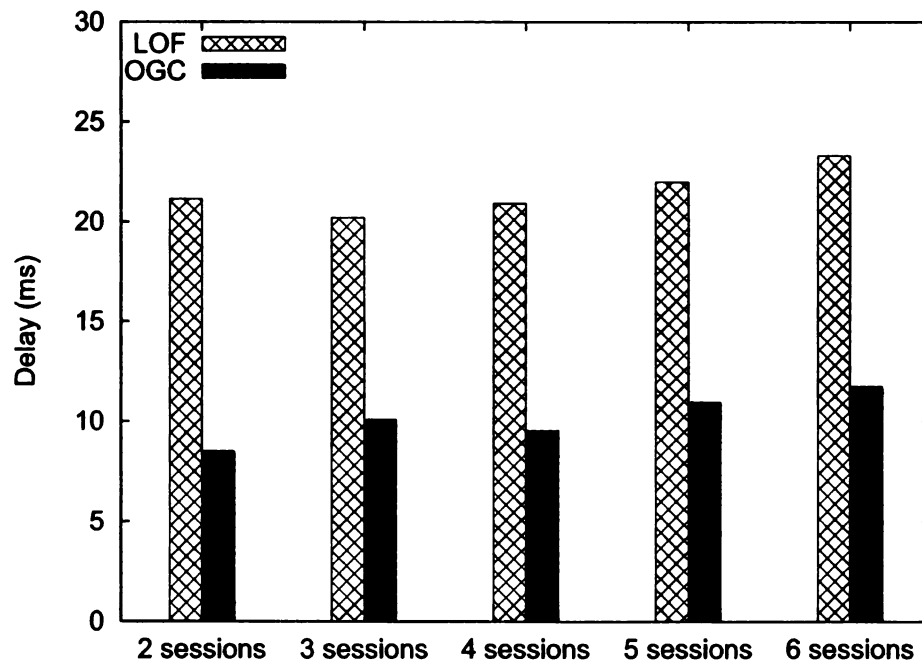


Figure 4.17. Average Delay vs. Number of Sessions (Multiple Channels)

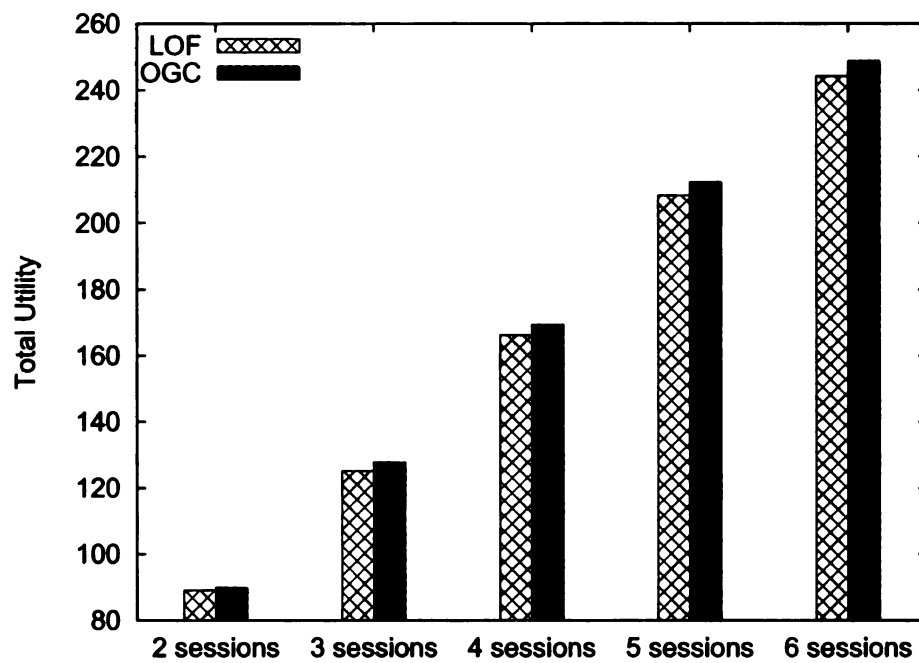


Figure 4.18. Total Utility vs. Number of Sessions (Multiple Channels)

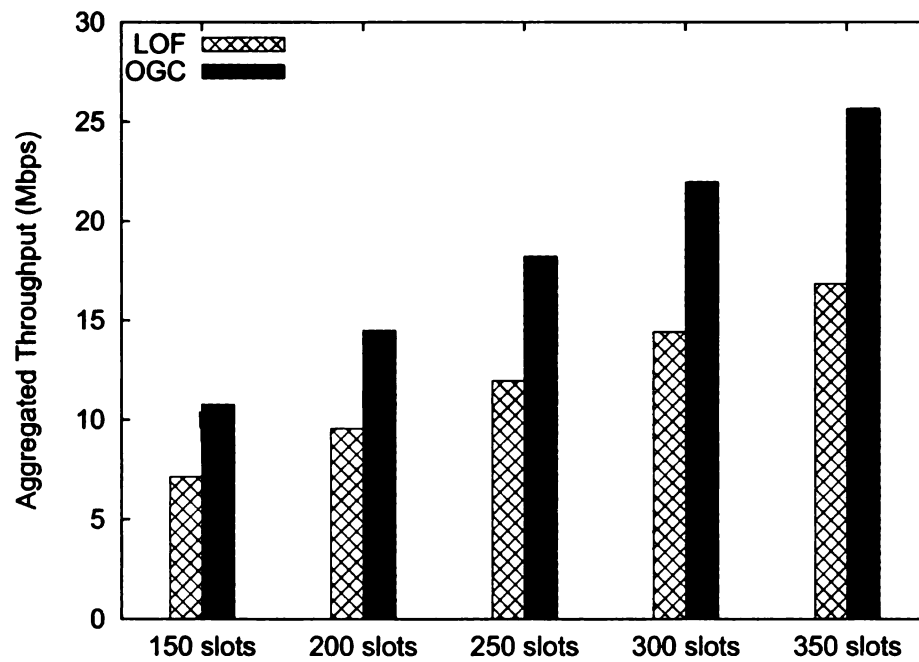


Figure 4.19. Aggregated Throughput vs. Number of Time Slots (Multiple Channels)

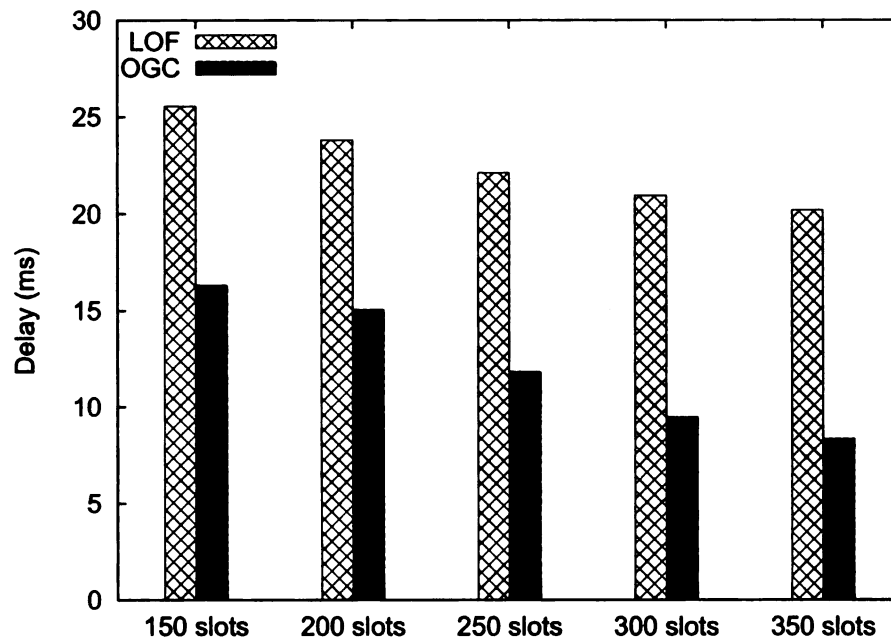


Figure 4.20. Average Delay vs. Number of Time Slots (Multiple Channels)

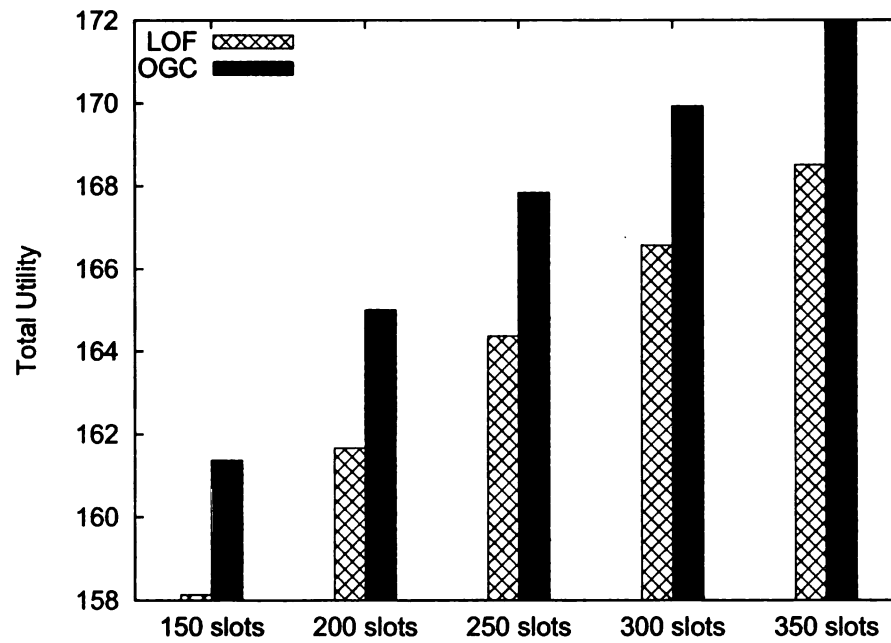


Figure 4.21. Total Utility vs. Number of Time Slots (Multiple Channels)

CHAPTER 5

Routing for Minimum Length

Schedule in Multi-channel TDMA

Based Wireless Mesh Networks

Multi-radio multi-channel communication is a promising technology to increase the network capacity. However, multiple radios may increase the hardware cost. Using a single radio with multiple channels can still significantly improve the network performance. IEEE 802.11b provides 3 orthogonal channels in the 2.4GHz spectral band and IEEE 802.11a provides 13 orthogonal channels in the 5GHz band [19]. With the IEEE 802.16 OFDMA [109] technology, the frequency band is divided into a large number of logical orthogonal sub-channels. For example, in [110], 30 logical sub-channels have been considered. In this chapter, we consider the network scenarios where each node is equipped with only one radio. Different links in the network are

assigned to different channels dynamically.

In this chapter, we discuss both routing and scheduling algorithms for multi-channel TDMA based wireless mesh networks. In accordance with the IEEE 802.16 (WiMAX) mesh network, we consider routing algorithms that build tree structures. The major focus of this chapter is to construct routing trees that yield schedules with minimum length. We try to solve two subproblems: 1) Find a TDMA schedule with minimum length, under the condition that the routing tree is given. 2) Construct the routing tree that results in the minimum length schedule.

Our work contains several major contributions. First, we propose a simple yet effective channel assignment algorithm to eliminate the secondary interference in the network, which enables us to use an optimal scheduling algorithm. Second, while providing significant performance improvement, our routing algorithm has low time complexity and is suitable for a centralized implementation.

5.1 Preliminaries

5.1.1 System Model

In this chapter, we only consider the aggregated traffic between the mesh routers and the gateway and we assume that mesh routers employ appropriate methods to distribute traffic to their clients. For simplicity, we only consider downlink traffic from the gateway node to mesh routers. However, our routing and scheduling algorithms can be applied to uplink traffic as well.

The network is modelled by a graph $G(V, E)$. V is the set of nodes in the network. An edge in set E indicates that two nodes are within transmission range of each other. Each node is equipped with a single radio, which can be tuned to different channels dynamically. We assume that there is a single gateway node in the network, denoted as g . We denote the neighbors of node $v \in V$ as $N(v)$. There are several traffic sessions that are from the gateway node to various mesh routers. The set of session is denoted as S and the set of session receivers is denoted as R .

A TDMA frame can be considered as a scheduling period, which consists of a fixed number of time slots. We denote this period as $[1, T]$, which consists of T time slots. The length of a schedule for supporting traffic requirements during a frame equals to the number of used time slots, i.e., slots during which at least one link is active.

For the ease of presentation, we define a packet as the element that can be sent in a time slot on the slowest link in the network. Since we consider periodic scheduling, we define the traffic requirement of a session receiver v as the number of packets to be received during a scheduling period. We denote the requirement as d_v . The capacity of a link (i, j) , denoted as c_{ij} , is defined as the number of packets that can be sent over the link in a single time slot. We also define the flow on a link (i, j) , denoted as f_{ij} , as the number of time slots needed to carry all the traffic sessions that are across this link during a scheduling period.

Interference can be classified into *primary interference* and *secondary interference*. Primary interference is due to the fact that a node cannot transmit and receive simultaneously. Secondary interference arises between two links when the receiver of one link is within the interference range of the sender of another link, or vice versa.

In this chapter, we assume that there are enough orthogonal channels available in the network to eliminate all the secondary interference. We propose a channel assignment algorithm to achieve this in Section 5.3.

Since we assume that secondary interferences are eliminated by using orthogonal channels, links only contend with each other if they share a common sender or receiver. According to the above contention model, we can define a *link contention (conflict) graph* $G_c(V_c, E_c)$ based on the contention relationship among different links. Contention graphs have been used in Chapter 3 and 4 to model network resource constraints. The vertex set V_c contains all the links in the network. An edge in set E_c indicates that two vertices (links) contend with each other.

5.1.2 Joint Routing and Scheduling Optimization

In this section, we establish a linear programming formulation of the joint routing and scheduling problem. To facilitate the formulation, we define the following variables:

$$x_{ij}^t = \begin{cases} 1, & \text{if link } (i, j) \text{ is active at time slot } t \\ 0, & \text{otherwise} \end{cases}$$

$$\pi_t = \begin{cases} 1, & \text{if time slot } t \text{ is used} \\ 0, & \text{otherwise} \end{cases}$$

$$y_{ij} = \begin{cases} 1, & \text{if node } i \text{ is node } j\text{'s parent in the routing tree} \\ 0, & \text{otherwise} \end{cases}$$

Given the above definition, we can define the joint routing and scheduling problem as the minimization of the set of used slots in the time period $[1, T]$:

$$\min \sum_{1 \leq t \leq T} \pi_t \quad (5.1)$$

subject to

$$\sum_{i \in N(v)} f_{iv} = d_v + \sum_{i \in N(v)} f_{vi}, \forall v \in V \setminus \{g\} \quad (5.2)$$

$$\sum_{i \in N(g)} f_{gi} = \sum_{i \in R} d_i \quad (5.3)$$

$$\sum_{i \in N(v)} x_{iv}^t + \sum_{i \in N(v)} x_{vi}^t \leq 1, \forall v \in V, \forall t \in [1, T] \quad (5.4)$$

$$f_{vi} \leq \sum_{t=1}^T x_{vi}^t \cdot c_{vi}, \forall i \in N(v), \forall v \in V \quad (5.5)$$

$$x_{vi}^t \leq y_{vi}, \forall i \in N(v), \forall v \in V, \forall t \in [1, T] \quad (5.6)$$

$$x_{vi}^t \leq \pi_t, \forall i \in N(v), \forall v \in V, \forall t \in [1, T] \quad (5.7)$$

$$\sum_{i \in N(g)} y_{ig} = 0 \quad (5.8)$$

$$\sum_{i \in N(v)} y_{iv} = 1, \forall v \in R \quad (5.9)$$

$$\sum_{i \in N(v)} y_{iv} \leq 1, \forall v \in N \setminus R \setminus \{g\} \quad (5.10)$$

Constraint 5.2 ensures the flow conservation in mesh routers. Constraint 5.3 ensures the flow conservation in the gateway node. Constraint 5.4 reflects the primary interference: links that are from or to a node cannot be active at the same time. Constraints 5.5, 5.6 and 5.7 bind the routing and scheduling: the flow on each link cannot exceed the total resource allocated to this link. Constraints 5.8, 5.9 and 5.10 ensure the routing tree structure.

Due to the high computational cost of solving the joint routing and scheduling problem, in this chapter we divide the overall problem into two subproblems:

1) Schedule the links in minimum number of time slots, under the condition that the routing tree and traffic requirements are given. We call this the link scheduling problem.

2) Given the network topology and traffic requirements, construct a routing tree that results in the shortest schedule length. We call this the routing problem.

In Section 5.1.3 we introduce an optimal scheduling algorithm based on graph coloring. In Section 5.2 we further refine the routing problem definition and propose a heuristic routing algorithm.

5.1.3 Optimal Link Scheduling

A scheduling problem can be translated into a graph coloring problem, in which time slots are considered as colors to be assigned to different links. In this chapter, we use the scheduling algorithm proposed in Chapter 4, which is based on greedy graph coloring with linear time complexity.

In [102], the authors proved that for a tree-structured wireless mesh network without secondary interference, its contention graph is a chordal graph. It has been shown that the greedy graph coloring is optimal for chordal graphs [106]. The optimal graph coloring uses minimum number of colors, so our scheduling algorithm produces schedules with minimum length.

5.2 Routing for Minimum Length Schedules

Given the routing tree and traffic requirement, our scheduling algorithm yields the optimal schedule with the minimum length. In fact, the length of the schedule equals to the size of the maximum clique of the extended contention graph G_e . A detailed discussion on the extended contention graph has been given in Section 4.2.3.

Since we ignore the secondary interference when we construct the contention graph G_c , each set of links that associate with a node v forms a maximal clique. We denote this set as $L(v)$. Based on the routing tree and the traffic requirements, we can calculate the flow (the number of time slots needed) on each link l . We denote this value as f_l . We define the cost of node v , $C(v)$, as follows:

$$C(v) = \sum_{l \in L(v)} f_l$$

According to the construction of G_e from G_c , each $C(v)$ corresponds to the size of a maximal clique of G_e . So the size of the maximum clique of G_e , or the length of the schedule is

$$\max_{v \in V} C(v)$$

The purpose of the routing algorithm is to minimize the above value. In fact, this problem is defined as the “minimization of the maximum nodal degree” [111]. It is a difficult combinatorial problem for which no known polynomial solution exists [111]. In this section we develop a heuristic algorithm to find suboptimal solutions.

Figure 5.1 shows an example scenario to illustrate the problem. In this figure, dashed lines depict links. For this example scenario, there are four traffic sessions: sessions from gateway node 0 to nodes 1, 2, 3 and 4 respectively. We specify the

capacity of each link in terms of number of packets per time slot. Assume that the rate requirement of each session is 12 packets per scheduling period.

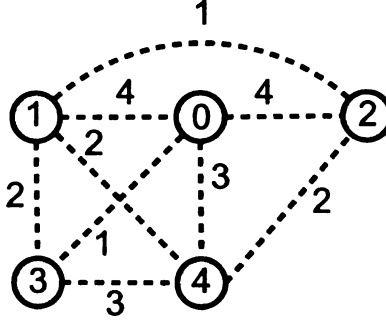


Figure 5.1. An Example Scenario

Figures 5.2(a) and (b) show the routing trees obtained by Breadth First Search (BFS) and Dijkstra algorithm [112]. In the Dijkstra algorithm, the weight of a link is set to be the inverse of the link capacity. Figure 5.2(c) shows the optimal routing tree. On each link, we show its flow value. As a result, we can calculate that 22 time slots are needed if we use BFS routing tree, and 14 time slots are needed if we use Dijkstra routing tree. The minimum number of time slots needed is 13 time slots with the optimal routing tree. From the above example we can see that the routing tree has a significant impact on the length of the TDMA schedule. Our routing algorithm aims to find the optimal routing tree that enables minimum length schedule.

5.2.1 Heuristic Routing Algorithms

First we introduce a modified version of the Dijkstra algorithm. We set the weight of each link to be the inverse of the link capacity and construct a spanning tree with the gateway as the root node. In the original Dijkstra algorithm, the distance between a

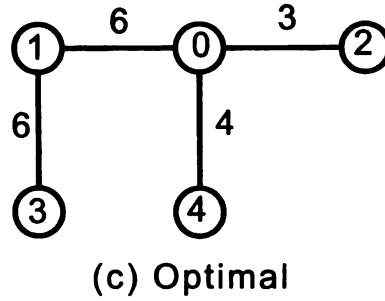
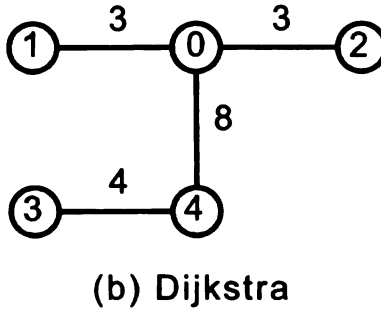
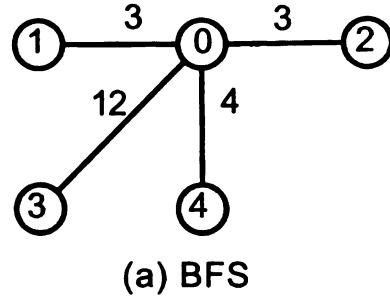


Figure 5.2. Routing Trees

vertex and the root node is defined as the sum of the weights of the links along the path. In our modified version of Dijkstra algorithm, the distance between a vertex and the root node is defined as the maximum weight of the nodes along the path. The weight of a nodes is defined as the sum of the weights of all the links associated with it. We get a spanning tree as the output of the modified Dijkstra algorithm. In fact, the path for each session, from the gateway to the session receiver, is optimal (which yields minimum length schedule) if it is the only session in the network.

Algorithm 5: Routing

input : network graph G , session set S
output: routing tree T

- 1 Define C as the vector of costs of nodes
- 2 Initialize the cost of each node to 0
- 3 Initialize the routing tree T to be empty
- 4 Run modified Dijkstra algorithm
- 5 Sort sessions in descending order according to path cost
- 6 Add the path with highest cost to the routing tree
- 7 **for** *each subsequent session* s **do**
 - 8 r = receiver of session s
 - 9 Calculate the weight of each link; the weight equals to the flow on the link
 - 10 $\text{Path}(g, r, C)$
 - 11 Add the path to the routing tree T
 - 12 Update C according to the new routing tree and the traffic requirements

The major idea of our routing algorithm is to construct the path for each session one by one. We have defined the cost of a node before. Here we define the cost of a path to be the maximum cost of nodes along this path. At the beginning, the routing tree is empty. We run the modified Dijkstra algorithm to obtain a spanning tree. We then calculate the cost of the path for each session assuming that it is the only session in the network. We sort the sessions by the costs of their paths in descending order. The path with the highest cost is added to the routing tree. We then run a path construction algorithm for other sessions following the descending order.

The path construction algorithm is similar to the modified version of the Dijkstra algorithm. The major difference is that the distance between a vertex and the root node is the cost of the path from the root node to the vertex. After each path construction, the cost of nodes along the path is updated and will be used in calculating the path cost when we construct the next path. The main algorithm is presented in

Algorithm 6: Path

input : sender s , receiver r , cost vector C
output: constructed path from s to r

- 1 Define vector t where $t[i]$ indicates if node i has been selected in the tree
- 2 Define vector d where $d[i]$ indicates the distance (path cost) from s to i
- 3 Define vector p where $p[i]$ indicates the parent node of node i

// initialize the vectors

- 4 **for each node** i **do**
 - 5 $t[i] = false$
 - 6 $d[i] = \infty$
 - 7 $p[i] = -1$
- 8 $d[s] = 0$
- 9 $v = s$ // start from the root node
- 10 **while** $t[v] == false$ **do**
 - 11 $t[v] = true$ // add this node to the tree
 - 12 **if** $v == s$ **then**
 - 13 **for each neighbor** i **of node** v **do**
 - 14 $w = \text{weight of edge } v \rightarrow i$
 - 15 // update the distance
 - 16 **if** $d[i] > d[v] + w + C[v]$ **then**
 - 17 $d[i] = d[v] + w + C[v]$
 - 18 $p[i] = v$
 - 19 **else**
 - 20 **for each neighbor** i **of node** v **do**
 - 21 $temp = 0$
 - 22 $w_1 = \text{weight of edge } p[v] \rightarrow v$
 - 23 $w_2 = \text{weight of edge } v \rightarrow i$
 - 24 **if** $w_1 + w_2 + C[v] > d[v]$ **then**
 - 25 $temp = w_1 + w_2 + C[v]$
 - 26 **else**
 - 27 $temp = d[v]$
 - 28 // update the distance
 - 29 **if** $d[i] > temp$ **then**
 - 30 $d[i] = temp$
 - 31 $p[i] = v$
 - 32 Set v to the node with minimum distance value
 - 33 **if** $v == r$ **then**
 - 34 return the path from s to r

Algorithm 5, and the detailed path construction algorithm is presented in Algorithm 6. We name the algorithm as Modified Dijkstra for Path (MDP).

In Algorithm 5, when adding a path to the routing tree, we start from the receiver node to the root node. When a node already exists in the constructed routing tree, we ignore the rest of the path and use the existing tree. This is for ensuring the tree structure of the output of the algorithm. Using priority queue implementation, the time complexity of the Dijkstra algorithm is $O((m+n) \log n)$, where n is the number of nodes, and m is the number of edges in the graph [112]. So the time complexity of our algorithm is $O((m+n)s \log n)$, where s is the number of traffic sessions in the network.

5.2.2 Algorithm Illustration

We use an example to illustrate our algorithm. The scenario is shown in Figure 5.3. Node 0 is the gateway. Link capacity (in terms of number of packets per slot) is noted in the figure. There are two sessions: session 1 is from gateway to node 5; session 2 is from gateway to node 7. Both of the sessions have traffic requirement of 6 packets per scheduling period.

In Figure 5.4 we show the execution of the modified Dijkstra algorithm. We show each step when a new node is added to the spanning tree. The distance value of each node when it is added to the spanning tree is shown.

After the execution of the modified Dijkstra algorithm, we obtain the paths for session 1 and session 2. Session 1 has a higher cost path, so its path is added to

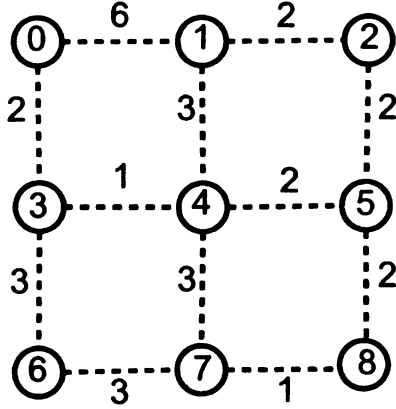


Figure 5.3. An Illustration Scenario

the routing tree, as shown in Figure 5.5(a). Figure 5.5(b) shows the execution of the Algorithm 6 to construct the path for session 2. We show each step when a new node is added to the tree. The distance value of each node when it is added to the tree is shown. Figure 5.5(c) shows the final routing tree and the cost value of each node.

5.3 Further Discussions

5.3.1 Channel Assignment

We propose a simple yet effective channel assignment algorithm to work with our scheduling and routing algorithms. The output of the scheduling algorithm indicates which group of links will be active simultaneously in each time slot. For each time slot, we need to assign channels to this group of links in order to eliminate secondary interference. The algorithm is described in Algorithm 7. A similar algorithm has been used in Chapter 4. Note that only secondary interference is possible among links that are active in the same time slot.

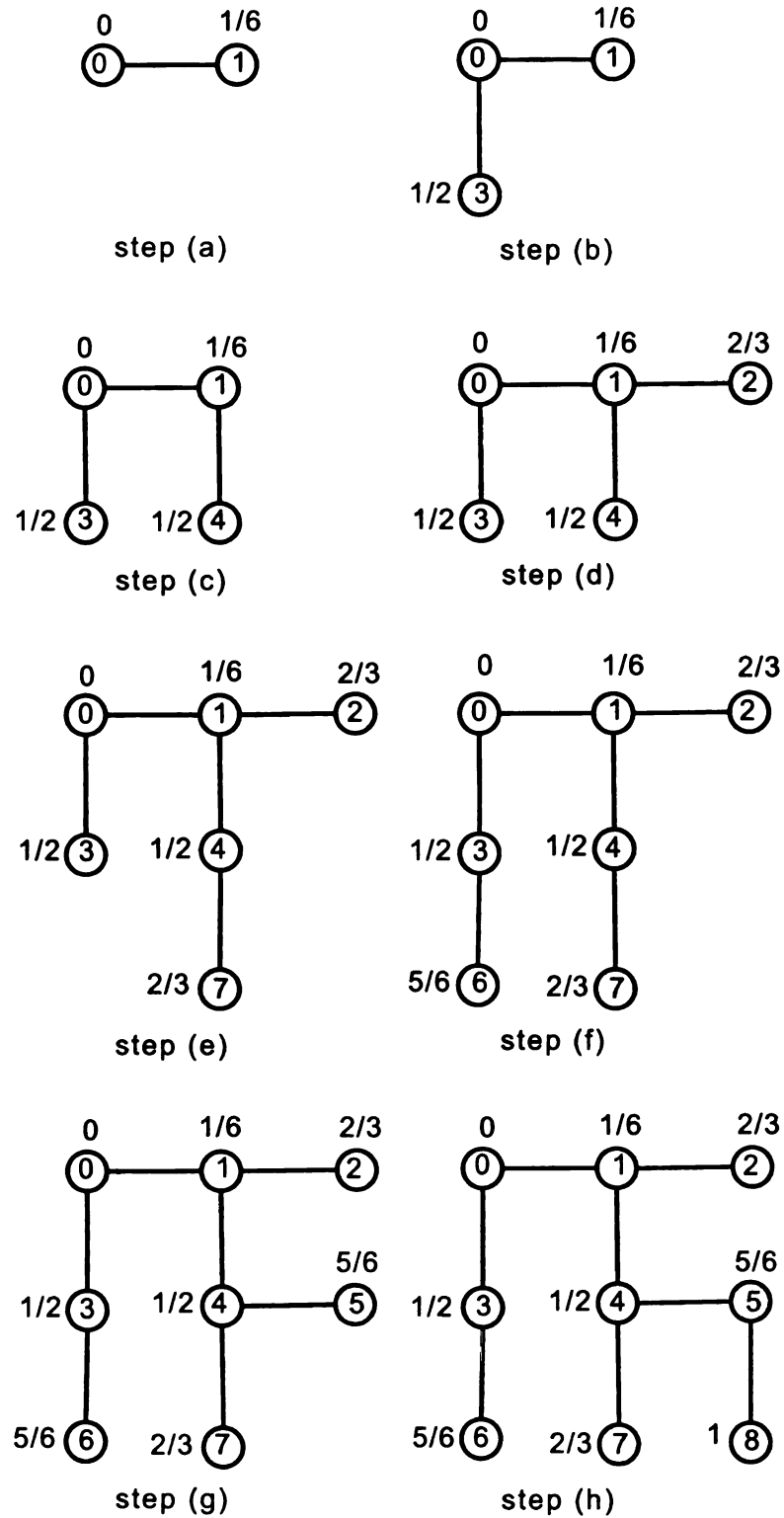


Figure 5.4. Modified Dijkstra Algorithm Illustration

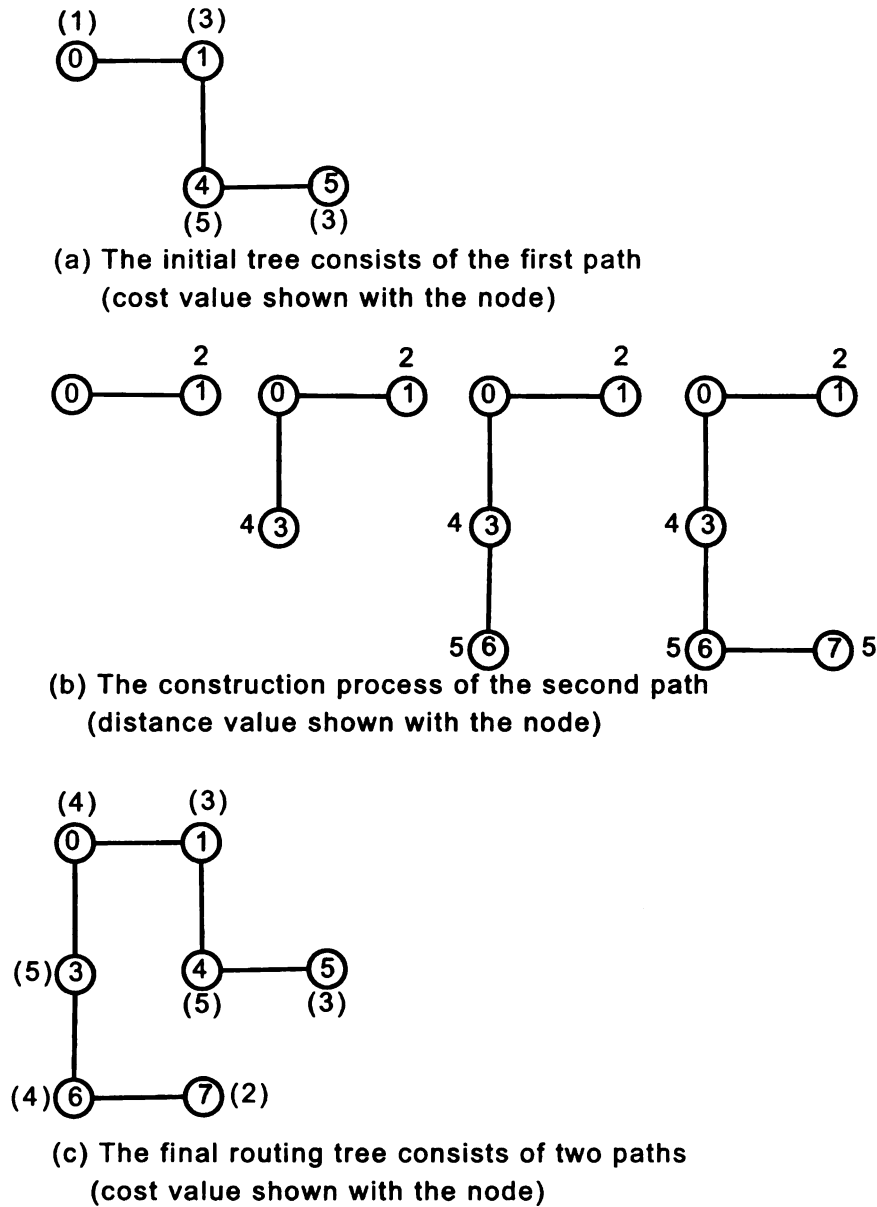


Figure 5.5. Routing Algorithm Illustration

5.3.2 Multiple Gateways

In this section, we discuss the algorithm extension when multiple gateways exist in the network. Multiple gateways are used to increase the network throughput and to balance the traffic load.

For each gateway, we can run the modified Dijkstra algorithm defined in Section

Algorithm 7: Channel Assignment

input : A set of links $L = [1, \dots, |L|]$
output: Channel to be used for each link $l \in L$
1 Define the vector of available channels as P
2 **for** $l = 1$ **to** $|L|$ **do**
3 $Q(l)$ = links in L that interfere with l
4 p = lowest channel in P that is not assigned in $Q(l)$
5 Assign channel p to link l

5.2 to construct a spanning tree. Each node in the network will get a distance value to each of the gateways. We associate a node to the gateway that gives minimum distance. Then for each gateway, we run the MDP algorithm defined in Section 5.2 for sessions whose receiver is associated to this gateway. Note that all the relay nodes used should also belong to the same gateway. After routing tree construction, the scheduling algorithm is performed separately at each gateway. Note that when we perform the channel assignment, the link group considered for each time slot should include links belonging to different routing trees.

5.4 Performance Evaluation

5.4.1 Simulation Setup

We use the ns-2 [107] simulator for performance evaluation. We simulate a flat area of 1000m by 1000m with randomly positioned stationary mesh routers. For each scenario we generate 30 random topologies, and our results represent the average performance over these 30 random topologies.

In our simulation, we use a TDMA MAC protocol similar to IEEE 802.16. A MAC

frame consists of a control subframe and a data subframe. Each subframe is divided into fixed-length time slots. The same frame structure is used in Chapter 4. The data subframe can be considered as the scheduling period. In the control subframe, mesh routers send rate requests to the gateway node via control message exchange. The gateway node is responsible for constructing the routing tree and establishing the schedule in a centralized manner. The routing and schedule information is then propagated to mesh routers during the control subframe. Since our routing and scheduling algorithms are very efficient, they can be executed when the network topology or the traffic requirement changes.

The physical and MAC layer simulation parameters are shown in Table 5.1. In our simulation, links have different rates. The link rate (capacity) is determined by the Signal to Noise Ratio (SNR) at the receiver side of the link. Table 5.2 lists different link rates [62] used in our simulation.

We establish CBR sessions between the gateway node and randomly chosen mesh routers. We set the packet size to be 48 bytes so that a single packet can be sent in a time slot using the lowest link rate. Table 5.2 also lists the link rate in the unit of packets/slot.

5.4.2 Simulation Results

We first put the gateway at the center of the topology. We conduct two sets of simulations. In the first set of simulation, we fix the number of mesh routers in the network to be 30 and vary the number of sessions in the network. In the second set

Table 5.1. Simulation Parameters

Parameter	Value
Frequency band (GHz)	5
Channel bandwidth (MHz)	20
Frame duration (ms)	20
Time slot duration (μs)	50
Transmit power (dBm)	25
Path Loss Model	$28.3 \log(d) + 41.9$, d is the distance
Noise Level (dBm)	$\text{bandwidth} * \frac{4 * 10^{-12}}{10^9}$

Table 5.2. Link Rate vs. SNR

Bit-rate (Mbps)	Capacity (pkts/slot)	SNR (dB)
23.04	3	8.5
46.08	6	15
69.12	9	21

of simulation, we fix the number of sessions to be 20 and vary the number of mesh routers in the network. The traffic rate of a session is randomly chosen in the set $\{90, 180, 270, 360, 450\}$ packets/second. Under this traffic rate distribution, the traffic requirements can be satisfied in all the scenarios.

We compare the number of time slots used by the scheduling algorithm under three different routing protocols: BFS, Dijkstra and MDP. Figure 5.6 and 5.7 show that MDP always enables the minimum length schedule. When we vary the number of sessions, the performance improvement of MDP over Dijkstra is between 20% and 30%. When we vary the number of mesh routers, the performance improvement is between 15% and 25%. We did not observe a meaningful trend of change in terms of performance gap when we vary the number of sessions or the number of mesh routers.

We then put the gateway node at the bottom-left corner of the network and

performed the same two sets of simulation mentioned above. The traffic rate of a session is randomly chosen from the set $\{90, 180, 270\}$ packets/second. Under this traffic rate distribution, the traffic requirements can be satisfied in all the scenarios. Figures 5.8 and 5.9 show that MDP always enables the minimum length schedule. When we vary the number of sessions, the performance improvement of MDP over Dijkstra is between 20% and 25%. When we vary the number of mesh routers, the performance improvement is between 10% and 15%. The performance gap is smaller compared with the scenarios where gateway is located at the center of the topology.

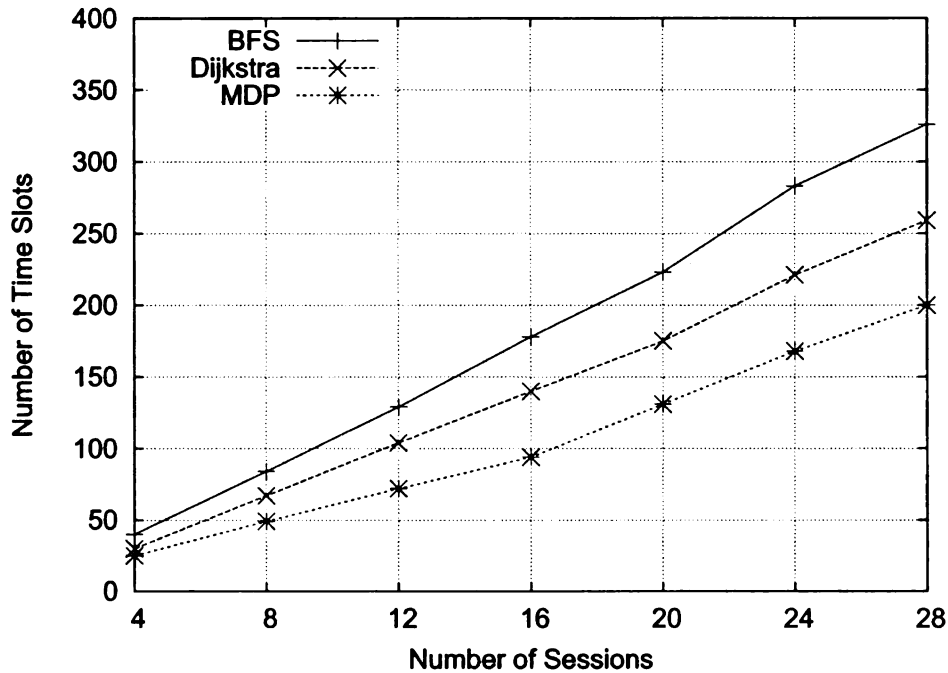


Figure 5.6. Number of Time Slots vs. Number of Sessions (Gateway at the Center)

We also implemented the channel assignment algorithm presented in Section 5.3 and measured the number of channels needed for eliminating secondary interference under all the simulated scenarios. We observed that the number of channels needed is always under 10.

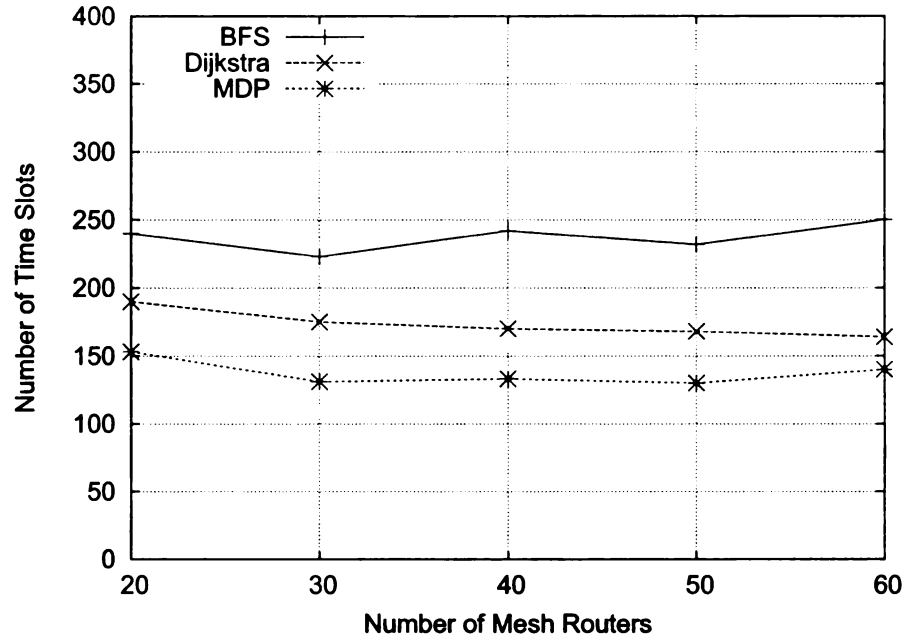


Figure 5.7. Number of Time Slots vs. Number of Mesh Routers (Gateway at the Center)

5.5 Summary

In this chapter, we considered the problem of routing and scheduling in multi-channel TDMA based wireless mesh networks. We proposed a channel assignment algorithm to eliminate secondary interference in the network using multiple orthogonal channels. With the channel assignment, we are able to use an optimal scheduling algorithm. We proposed a heuristic routing algorithm that tries to achieve minimum length schedule. Our algorithms have low time complexity and are suitable for centralized implementation. Our algorithms can be easily incorporated into WiMAX mesh networks.

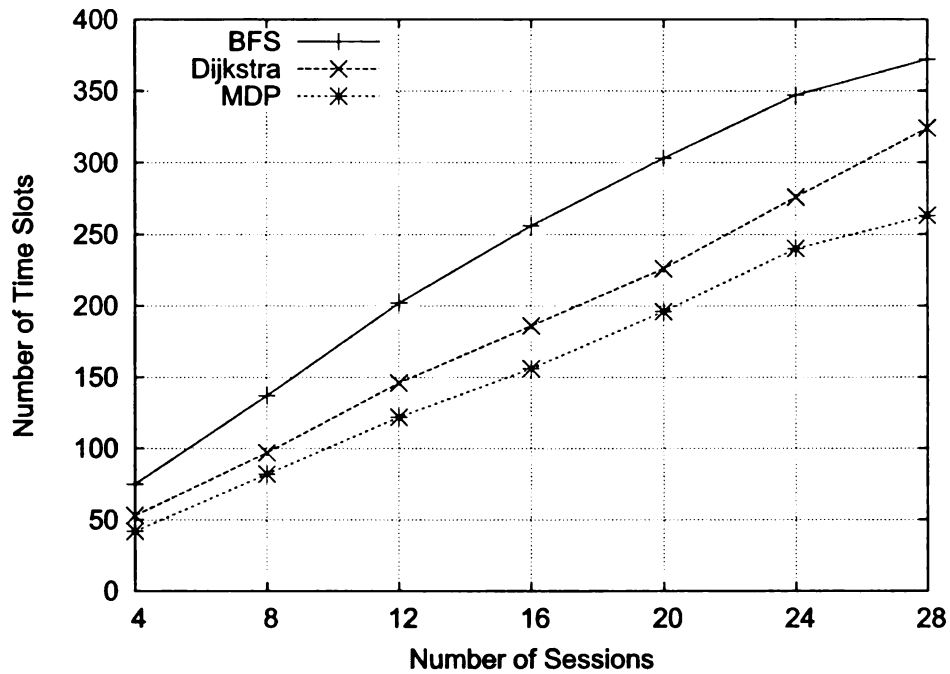


Figure 5.8. Number of Time Slots vs. Number of Sessions (Gateway at the Corner)

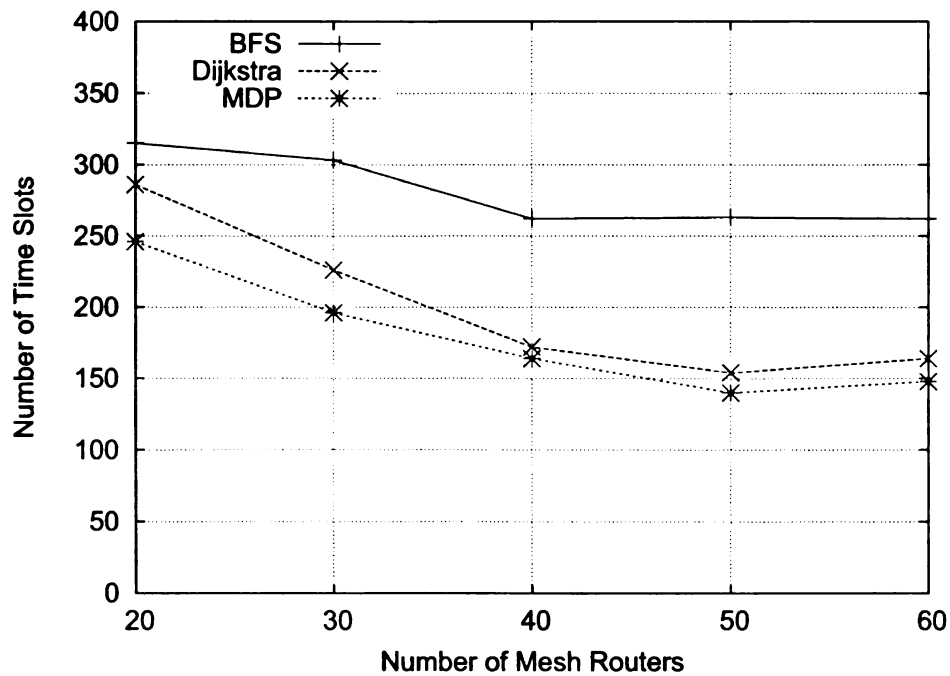


Figure 5.9. Number of Time Slots vs. Number of Mesh Routers (Gateway at the Corner)

CHAPTER 6

Path Selection for Mobile Stations in IEEE 802.16 Multihop Relay Networks

The path capacity or the maximum end-to-end throughput is a good metric to measure the quality of a path, however, it fails to reflect the achievable end-to-end throughput and delay of mobile stations in IEEE 802.16 MMR networks. In addition to path capacity, the achievable throughput of a mobile station in MMR network also depends on the available communication resources of a cell and the scheduling algorithm used in the base station. In this chapter, we propose a path selection metric, named Normalized Number of Minislots (NNM), which enables a mobile station to choose a path that satisfies its application rate and delay requirements. For simplicity, we only consider the downlink traffic from the base station to various mobile

stations. Our discussion on path selection only concerns the downlink routing tree. Our work can also be applied to path selection in the uplink routing tree.

6.1 Preliminaries

6.1.1 Frame Structure

According to the IEEE 802.16j standard draft, the MAC frame structure is based on the Orthogonal Frequency Division Multiple Access (OFDMA) transmission technique [109]. Under the Time Division Duplex (TDD) scheme, a MAC frame consists of a Downlink (DL) sub-frame period and an Uplink (UL) sub-frame period. Figure 6.1 shows the frame structure of a two-hop relay network. Some time intervals in the frame are reserved for control messages and contention-based communications.

OFDMA is a multi-carrier transmission scheme. Under OFDMA, the whole frequency band is segmented into a set of sub-carriers. Sub-carriers are grouped into logical entities called sub-channels. Sub-channels can be constructed in two different ways: grouping contiguous sub-carriers or grouping sub-carriers scattered over the entire frequency band. The second method is considered now by WiMAX vendors as the most relevant solution for practical wireless access systems [110]. It simplifies the resource allocation of a cell since all sub-channels are equally adequate for all users [110]. Under OFDMA, the time domain is segmented into constant symbol durations, thus the communication resources of a cell can be represented as a two-dimensional matrix. Each resource element is a combination of the symbol duration and the sub-

channel. Resource elements are allocated in repeated frame period to different links of the network.

In this chapter, we only consider the resource allocation in the time domain, i.e, the symbols are carried on all available sub-channels. In addition, symbol durations are grouped into minislots. A minislot contains a fixed number of symbol durations and is considered as the basic resource allocation unit in this chapter. Note that the above choice is made for the ease of presentation and simulation and is consistent with the Orthogonal Frequency Division Multiplexing (OFDM) transmission technique [109]. Our work can be easily extended to accommodate OFDMA transmission technique by changing the resource allocation granularity from minislot to the element of the two-dimensional matrix mentioned above.

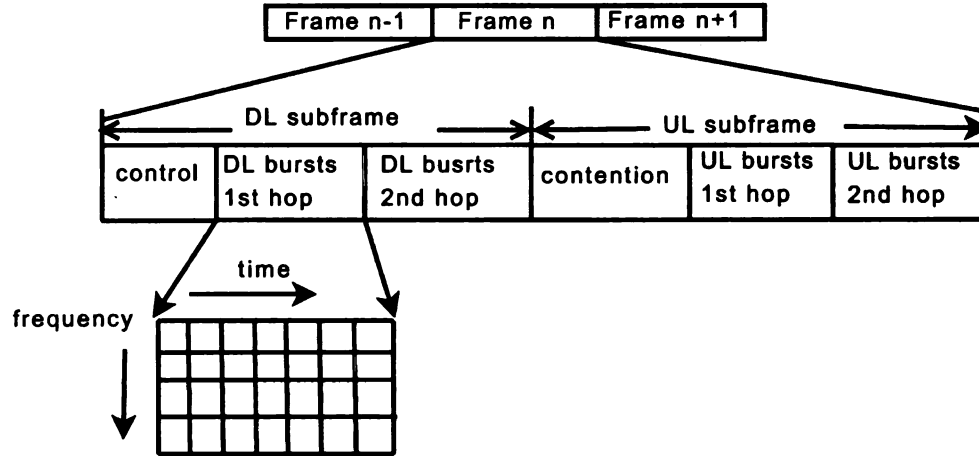


Figure 6.1. IEEE 802.16j MAC Frame Structure

6.1.2 Fast Access Station Switching

In this chapter, we assume that the fast access station switching [113] method is used. Using this method, a mobile station can change its access station from frame to frame

depending on the access station selection mechanism [9]. Thus, the path selection is performed before the scheduling algorithm at each frame by different base stations. In order to use this method, mobile stations and relay stations should periodically monitor the quality of its uplink and downlink, and report them to the associated base station. The link quality measurement and report mechanisms have been discussed in several contributed documents [114] to the IEEE 802.16's relay task group.

6.2 Path Selection Metric

In this section, we introduce three path selection metrics. The first one, “number of hops”, is very easy to implement. However it does not take the path capacity into consideration, which is one of the important factors that affect the achievable throughput and delay of the mobile stations. The second one, “maximum end-to-end throughput”, is based on the previous work in [79]. This metric reflects the path capacity. However, it does not consider the achievable throughput of a mobile station. Finally we discuss our proposed path selection metric, “normalized number of minislots”, which enables mobile stations to achieve high end-to-end throughput and low delay. The path selection metric can be used by the base station in a centralized manner or by mobile stations in a distributed manner.

Before discussing these path selection metrics, we introduce notations that will be used in this section. Denote the set of cells as $Q = \{q_1, \dots, q_M\}$, the set of mobile stations as $V = \{v_1, \dots, v_N\}$ and the set of links as $E = \{e_1, \dots, e_L\}$. Assume that the link rate, in bits per second, of link e_l is r_l . The link rate depends on the modulation

and coding scheme used for the link, which is determined by the Signal to Noise Ratio (SNR) at the receiver side of the link. Table 4.3 in Chapter 4 shows the type of modulation and link rate (represented as bits/symbol) in relation to the received SNR [62, 115].

6.2.1 Number of Hops (NOH)

The simplest path selection metric is the number of hops between the mobile station and the base station. Mobile stations choose the path with the smallest number of hops. This metric is widely used in several routing protocols for mobile ad hoc networks. The major drawback of this metric is that it usually results in long hops that have poor link rates.

6.2.2 Maximum End-to-End Throughput (MET)

In [79], the authors propose that the maximum end-to-end throughput of a given multi-hop path p_n may be expressed as

$$MET = \frac{b}{ETT} \quad (6.1)$$

where b is the number of bits per packet, ETT is the expected transmission time of a packet on path p_n . ETT has originally been proposed and discussed in [35] for IEEE 802.11 mesh networks.

ETT can be calculated as

$$ETT = \sum_{l: e_l \in p_n} ETT_l$$

where ETT_l is the expected transmission time of a packet on link e_l . For simplicity, we don't consider packet retransmissions caused by link errors. So ETT_l can be calculated as

$$ETT_l = \frac{b}{r_l}$$

To get the optimal throughput, a mobile station should choose a path with the maximum MET value.

6.2.3 Normalized Number of Minislots (NNM)

Equation (6.1) calculates the maximum throughput of a path, however, it does not correctly reflect the achievable end-to-end throughput of a mobile station when the IEEE 802.16 MAC frame structure and scheduling algorithms are taken into consideration. In the following, we calculate the end-to-end throughput of a mobile station v_n . Assume that the mobile station uses path p_n , which belongs to cell q_m . Denote the traffic rate of mobile node v_n , in bits per second, as g_n . Denote the throughput of the mobile node, in bits per second, as \hat{g}_n .

The throughput of mobile station v_n can be calculated as [62]

$$\hat{g}_n = \min_{l: e_l \in p_n} \frac{\hat{s}_l}{s_l} g_n$$

where s_l is the number of required minislots on link e_l . \hat{s}_l is the number of assigned minislots on link e_l . As explained in [62], the achievable end-to-end throughput is the minimum bandwidth the mobile station gets on all the links on the path.

In fact,

$$s_l = \sum_{n: p_n \ni e_l} \left\lceil \frac{g_n}{r_l t} \right\rceil$$

where t is the duration of a minislots. Here is a brief explanation of the above equation. Link e_l carries traffic of several mobile stations whose path contains the link. For every mobile station, the required transmission time on link e_l equals to its traffic rate divided by the link rate. Thus, for every mobile station, the number of required minislots on link e_l equals to the required transmission time divided by the duration of a minislots. Finally, the total number of required minislots on link e_l is the sum of required minislots of all mobile stations which uses path that contains the link.

The number of assigned minislots on link e_l depends on the number of available minislots in a frame for downlink traffic in the cell and the scheduling algorithm used in the base station. In this chapter, we use a simple IEEE 802.16 scheduling algorithm presented in Section 2.3.2. For this scheduling algorithm,

$$\frac{\hat{s}_l}{s_l} = \min \left\{ 1, \frac{S_m}{\sum_{l:e_l \in q_m} s_l} \right\}$$

where S_m is the number of available minislots in a frame for downlink traffic in the cell q_m to which the path p_n belongs. $\sum_{l:e_l \in q_m} s_l$ represents the total number of minislots required for all links in the cell q_m .

Similar to [65], we denote \hat{s}_l/s_l as the *downlink satisfaction index* of link e_l . In fact, the satisfaction index represents the proportion of the number of assigned minislots over the number of required minislots. It attains maximum value of 1 when the number of assigned minislots equals to the number of required minislots. According to the scheduling algorithm, all links that belongs to the same cell have the same downlink satisfaction index, so we can define the *downlink satisfaction index* of the

cell q_m as

$$SI_m = \min \left\{ 1, \frac{S_m}{\sum_{l: e_l \in q_m} s_l} \right\}$$

We also define the *downlink load ratio* of the cell q_m as

$$LR_m = \frac{\sum_{l: e_l \in q_m} s_l}{S_m}$$

Thus the end-to-end throughput of mobile station v_n can be expressed as

$$\hat{g}_n = SI_m \cdot g_n$$

When the number of available minislots is larger than the total number of required minislots, the cell is underloaded. In this case, all links are assigned the number of minislots according to their requirements, and mobile station's throughput equal to its traffic rate since the downlink satisfaction index is 1. In addition, the delay of a packet is within the length of a frame. When the number of available minislots is smaller than the total number of required minislots, the cell is overloaded. In this case, all links are assigned a number of minislots that is scaled down according to the downlink load ratio in order to make sure that all assigned minislots fit into a single frame. In this case, a mobile station's throughput will be smaller than its traffic rate since the downlink satisfaction index is smaller than 1. More importantly, some packets that sent out in a frame can only be delivered to the destination in subsequent frames. This makes the delay of packets to be of several frame durations. From the above discussion we can see that the overloading of cells reduces the performance of mobile stations. The increased delay has especially detrimental effects on real-time multimedia applications.

In order to reduce the chances that cells get overloaded, we define the following minimization problem,

$$\min \sum_{m=1}^M LR_m \quad (6.2)$$

The solution to this minimization problem, which will be shown in the following, can be translated into a path selection metric. In fact,

$$\begin{aligned} \sum_{m=1}^M LR_m &= \sum_{m=1}^M \frac{\sum_{l:e_l \in q_m} s_l}{S_m} \\ &= \sum_{n=1}^N \frac{\sum_{l:e_l \in p_n} \lceil \frac{q_n}{r_l t} \rceil}{S_{p_n}} \end{aligned}$$

where S_{p_n} is the number of available minislots in the cell to which path p_n belongs.

The last equality is based on the fact that the number of required minislots on all links in a cell equals to the number of required minislots to support traffic of mobile stations in this cell. Based on the above analysis, we define a metric named NNM of a path p_n as follows,

$$NNM = \frac{\sum_{l:e_l \in p_n} \lceil \frac{q_n}{r_l t} \rceil}{S_{p_n}}$$

NNM equals to the number of minislots that will be used by mobile node v_n using path p_n divided by the total number of available minislots for downlink traffic in the cell to which p_n belongs. In order to solve the minimization problem defined in (6.2), mobile stations should use a path that has the minimum NNM value.

Note that the number of available minislots in a frame for downlink traffic is a configurable and adaptive parameter of a cell. It depends on the number of minislots needed for control messages and contention-based communications in a frame and the downlink-to-uplink-subframe ratio [109], which can be dynamically adjusted based on

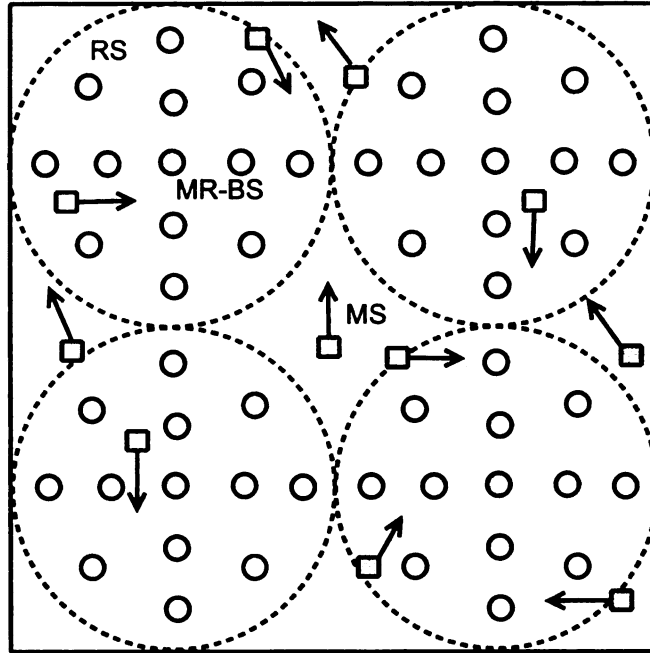


Figure 6.2. Simulation Topology

the network traffic profile.

6.3 Performance Evaluation

6.3.1 Simulation Setup

We use the ns-2 [107] simulator for the performance evaluation. The simulated network topology is shown in Figure 6.2. It is a 4000m by 4000m flat area with four cells. The radio coverage areas of base stations are shown in dotted circles. The shaded circles represent base stations; unshaded circles represents relay stations; shaded square represents mobile stations. Each cell has one base station and 12 associated relay stations. In each cell, the distance between the base station and the relay stations located near the center of the cell is 250m. The distance between the base station

and the relay stations located near the edge of the cell is 700m. The base station establishes a one-hop path with all associated relay stations. We assume that different cells use different frequency bands, so there is no inter-cell interference in our simulation. Other detailed simulation parameters are given in Table 6.1.

Table 6.1. Simulation Parameters

Parameter	Value
Frequency band (GHz)	5
Channel bandwidth (MHz)	20
Frame duration (ms)	20
OFDM symbol duration (μ s)	12.5
Minislot duration (ms)	0.1
Minislots in a frame	200
Minislots for downlink	[40,160]
MAC PDU (bytes)	96
RS queue size (packets)	100
BS transmit power (dBm)	30
RS transmit power (dBm)	25
Path Loss Model	$28.3 \log(d) + 41.9$, d is the distance
Noise Level (dBm)	$\text{bandwidth} * \frac{4 * 10^{-12}}{10^9}$, according to [108]

The number of mobile stations is varied from 20 to 60 in different simulation scenarios. Mobile stations' positions are randomly generated. We use the random way-point mobility model. Each mobile station picks a direction and a speed randomly. After reaching the destination, the mobile station pauses for some time and moves to a new direction with a new speed. In our simulation, the speeds of mobile stations are uniformly distributed in the range [1m/s, 30m/s]. The pause times are uniformly distributed in the range [0s, 10s].

All base stations are connected to a fixed node, denoted as the content station,

through a wired link. One CBR connection is established between the content station and each mobile station. The CBR traffic rate is varied from 200Kbps to 600Kbps in different simulation scenarios. The packet size is set to be 96 bytes. We use the average throughput and average delay of all mobile stations as the metrics to evaluate the performance of different path selection metrics. We also present the average downlink satisfaction index and the average downlink load ratio of the cells over all frames during the simulation.

6.3.2 Simulation Results

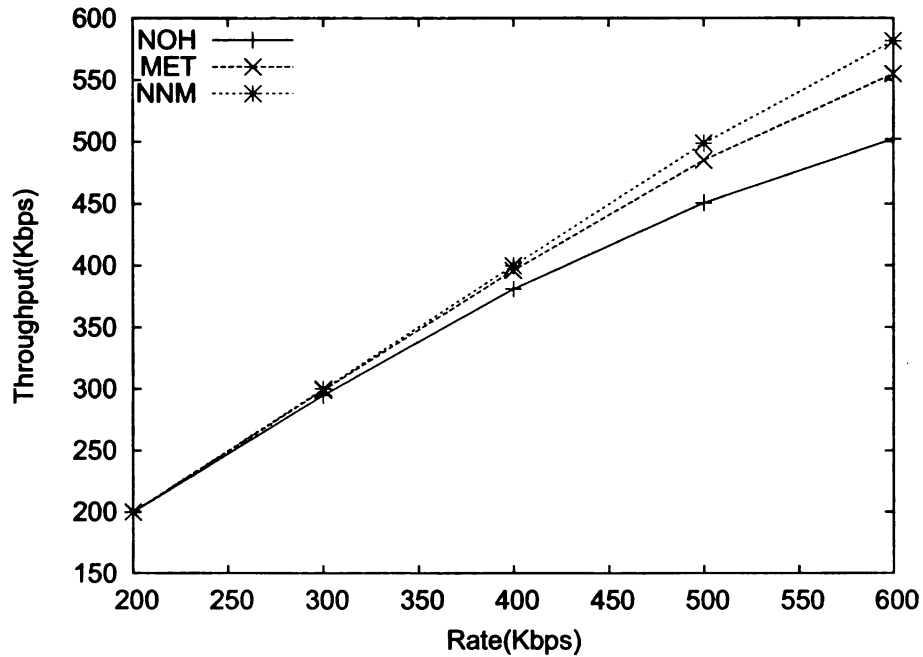


Figure 6.3. Throughput vs. Traffic Rate

We first vary the CBR traffic rate from 200Kbps to 600Kbps. In this set of simulations, we fix the number of mobile stations in the network to be 40. For every traffic rate, we conducted 10 simulation runs with different mobility scenarios and

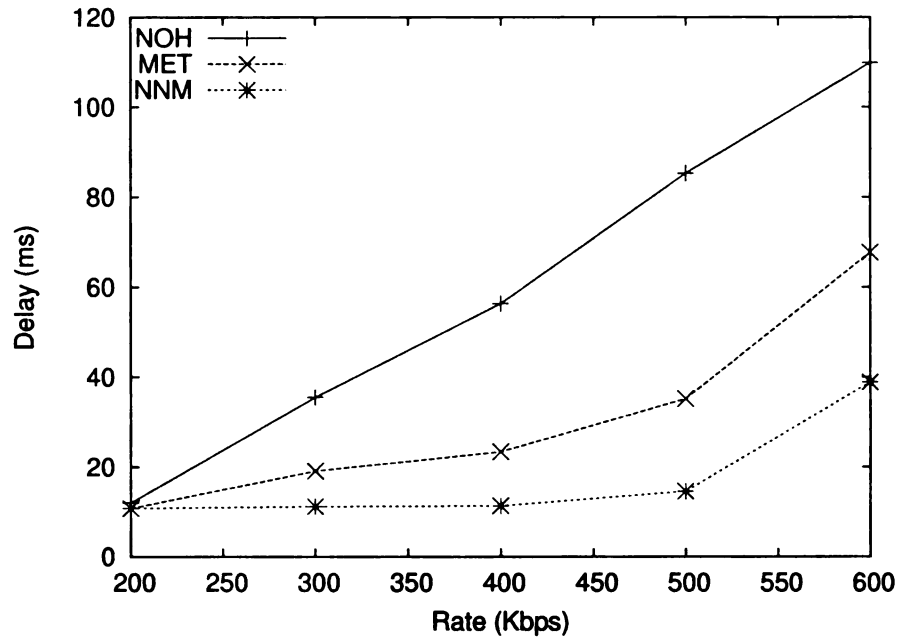


Figure 6.4. Delay vs. Traffic Rate

different numbers of available downlink minislots for cells. The throughput and delay value presented in Figures 6.3 and 6.4 are average values of all mobile stations in 10 simulation runs. Each simulation run lasts for 60 seconds.

Figure 6.3 shows the average throughput of mobile stations using three path selection metrics under different traffic rates. We observe that for all traffic rates, NNM achieves the highest throughput while NOH gives the lowest throughput. Figure 6.4 shows the average delay of mobile stations using three path selection metrics under different traffic rates. NNM achieves lowest delay for all traffic rates.

When the traffic rate is low, the throughput equals to the traffic rate and the delay is within a single frame duration. From Figure 6.5, we can see that the average satisfaction index equals 1 when the traffic rate is low, which means no cell is overloaded. When the traffic rate increases, the performance gain of using NNM becomes

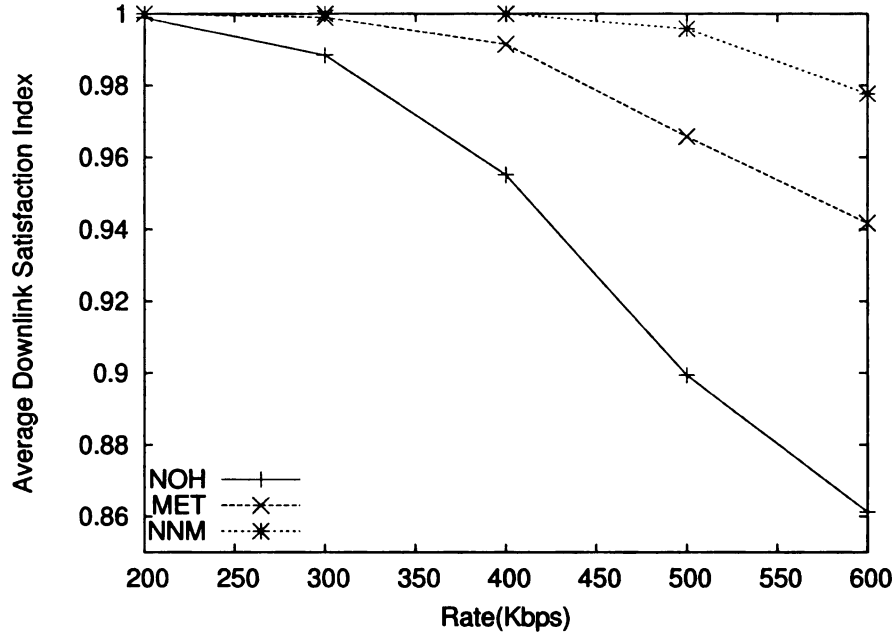


Figure 6.5. Downlink Satisfaction Index vs. Traffic Rate

more obvious. With NOH and MET, some cells become severely overloaded, which results in degraded throughput and delay. On the other hand, even with a very high traffic rate, NNM enable cells to maintain high downlink satisfaction index, which result in better throughput and delay performance for mobile stations compared with NOH and MET. We also observe that the performance gain of using NNM in terms of delay is much higher than throughput. This shows that the overloading of the cell has a much higher impact on delay than on throughput. From Figure 6.6 we can see that NNM yields minimum average downlink load ratio of cells for all traffic rates.

We also conduct simulations by varying the number of mobile stations (the density of mobile stations) in the network while fixing the traffic rate at 400Kbps. For each density, we conducted 5 simulation runs with different mobility scenarios and different numbers of available downlink minislots for cells. The throughput and delay

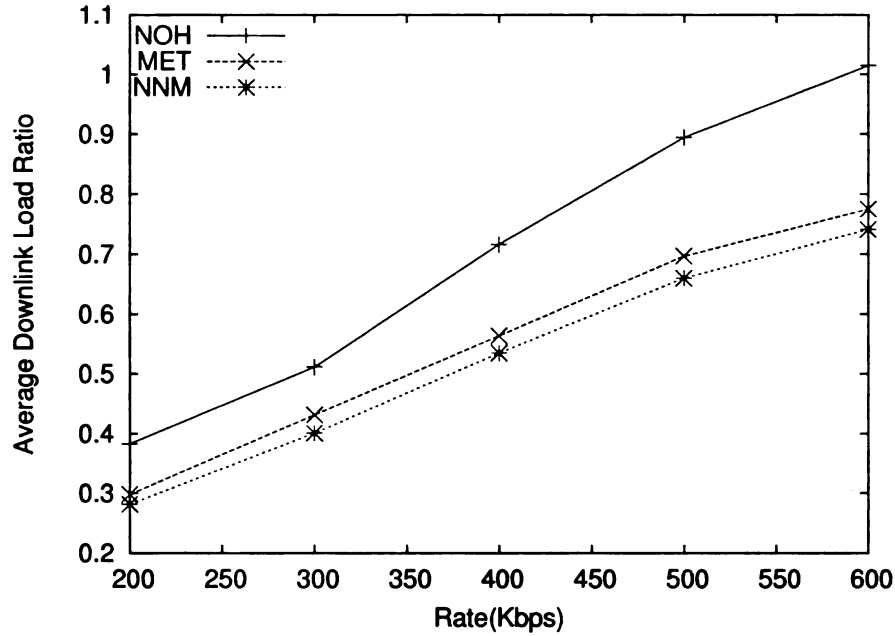


Figure 6.6. Downlink Load Ratio vs. Traffic Rate

value presented in Figures 6.7 and 6.8 are average values of all mobile stations in 5 simulation runs. Each simulation run lasts for 60 seconds.

Figures 6.7 and 6.8 show the average throughput and delay of mobile stations using three path selection metrics under different mobile station densities. Figures 6.9 and 6.10 show the average downlink satisfaction index and downlink load ratio of cells. We observe that for all densities, NNM gives the best performance while NOH always gives the worst performance. Increasing the number of mobile stations in the network has the same effect of increasing the traffic rate: the downlink traffic load of the network increases. So we observe similar performance gain of NNM compared to NOH and MET as in the simulation set while the traffic rate is varied.

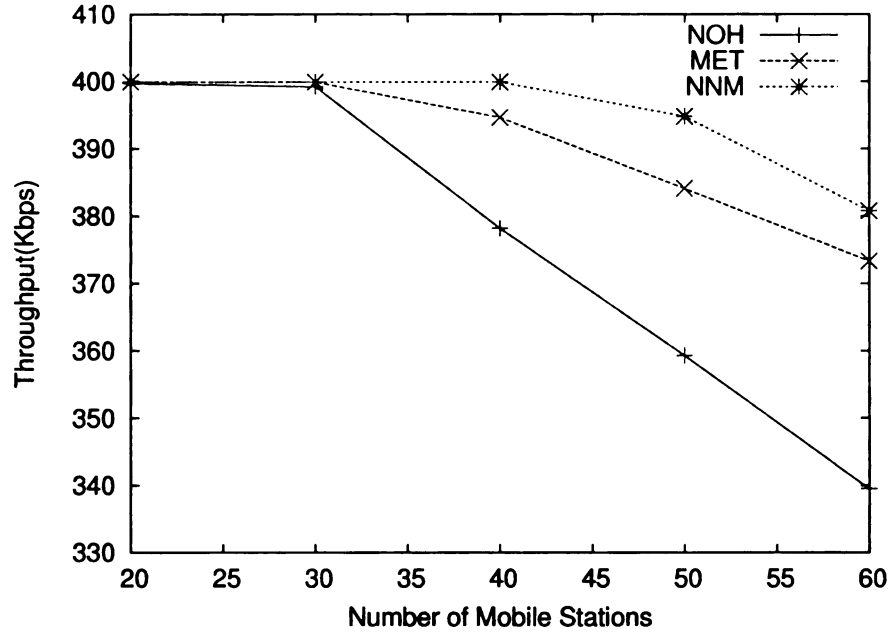


Figure 6.7. Throughput vs. Number of Mobile Stations

6.4 Summary

We have proposed a path selection metric for mobile stations in IEEE 802.16 MMR networks to select appropriate paths to the base station. Based on the calculation of the end-to-end throughput of mobile stations, we observed that the overloading of the cells has detrimental effects on the QoS of mobile stations, especially for real-time multimedia applications. Based on the above observation, we designed the path selection metric NNM, which enables the minimization of the sum of downlink load ratios of cells in the networks. Our proposed path metric effectively takes all the factors that affect end-to-end achievable throughput into consideration. Simulation results demonstrated that our metric yields better performance in terms of throughput and delay compared to existing path selection metrics, especially when networks have high traffic load.

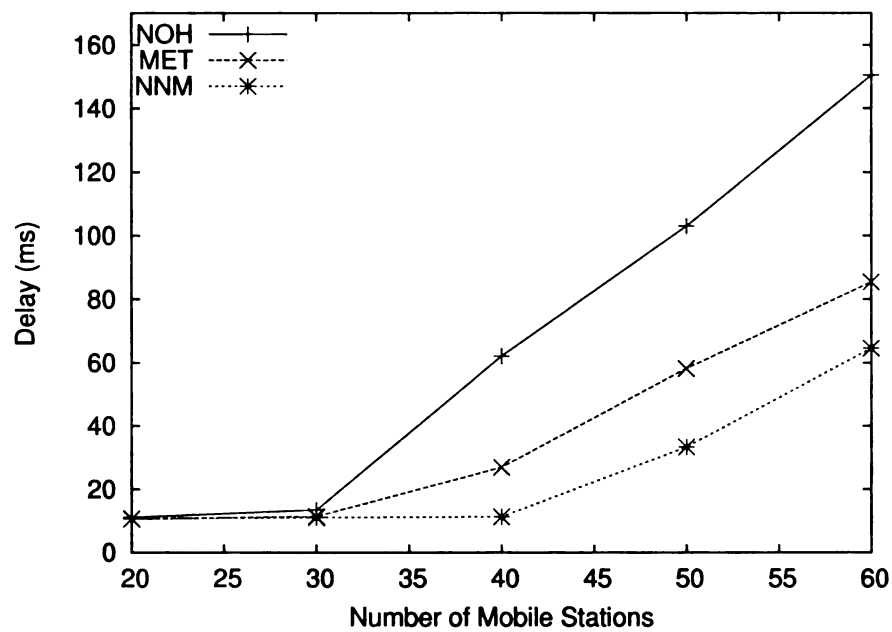


Figure 6.8. Delay vs. Number of Mobile Stations

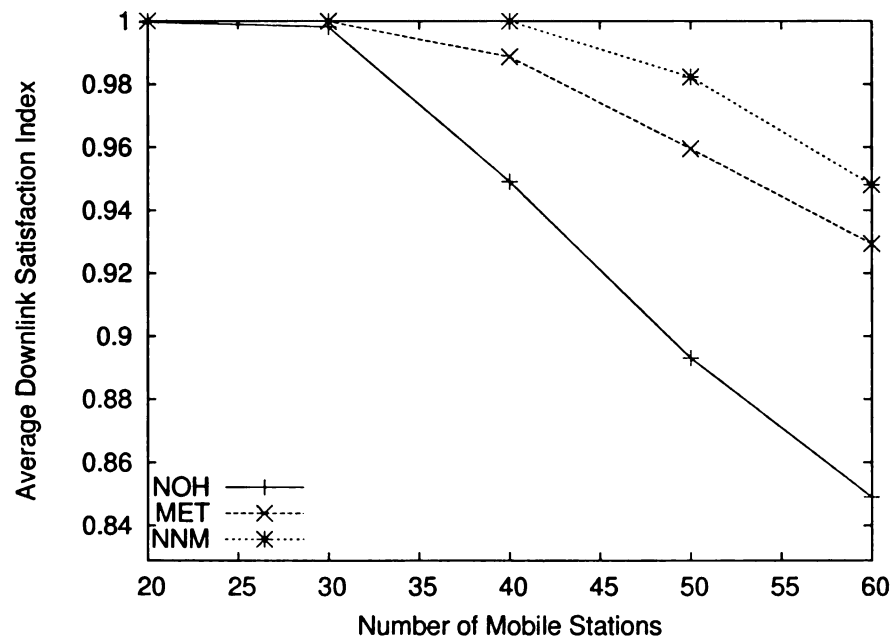


Figure 6.9. Downlink Satisfaction Index vs. Number of Mobile Stations

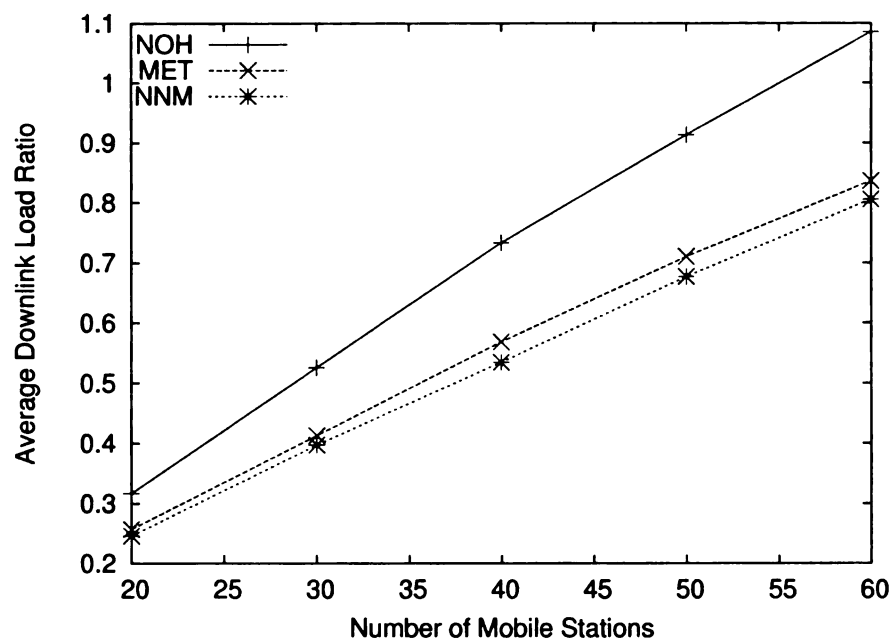


Figure 6.10. Downlink Load Ratio vs. Number of Mobile Stations

CHAPTER 7

Conclusions and Future Work

7.1 Conclusions

Wireless mesh networks become popular for supporting the last-mile broadband access. However, there are several challenges for the wide deployment of wireless mesh networks. The first one is the fair and efficient rate allocation. CSMA/CA MAC protocol coupled with TCP rate control mechanism results in severe unfairness. Users that are farther away from the Internet gateway receive less bandwidth and are sometimes starved. The second challenge is the QoS support of real-time applications. Multimedia applications over wireless networks become more and more popular and they require guaranteed throughput and low delay.

In this dissertation, we discussed several research topics related to rate allocation and QoS support in wireless mesh networks. For CSMA/CA based wireless mesh networks, we proposed a framework that combines fair rate allocation for elastic traffic and QoS support for real-time traffic. Our framework gives high priority to real-time

traffic by reserving necessary bandwidth for them. The left-over bandwidth is allocated fairly to elastic traffic through the Network Utility Maximization framework. We also discussed the centralized implementation and practical issues of the proposed framework. A centralized implementation provides more accurate bandwidth reservation for real-time traffic and faster, more stable rate allocation for elastic traffic.

With the increasing popularity of multicast applications, it is important to consider networks with both multicast and unicast traffic. Multicast is helpful for bandwidth sharing, but it should not starve the unicast traffic. We proposed a rate allocation framework for multicast and unicast traffic in TDMA based wireless mesh networks. In addition, admission control and bandwidth reservation of real-time traffic has been incorporated into our framework. Using a TDMA based MAC protocol facilitates more efficient rate allocation and more flexible QoS support. We extended the notion of “link contention graph” to “transmission contention graph” in order to model the rate allocation constraints. By using receiver-oriented utility functions and the NUM framework, our rate allocation provides efficient and fair rate allocation to both unicast and multicast traffic. We also proposed a scheduling algorithm based on graph coloring to realize the allocated rates. Our proposed framework has low time complexity and can be easily incorporated with WiMAX mesh networks.

In TDMA based wireless mesh networks, routing and scheduling algorithms are essential to provide QoS support for mesh clients. In order to maximize the network throughput and minimize the session delay, the routing and scheduling algorithms should produce a minimum length schedule. We proposed a heuristic routing algorithm that aims to build the routing tree that results in the minimum length schedule.

The work discussed above only concerns the static mesh network backhaul. The introduction of mobile stations to mesh networks brings new challenges for QoS support. To support multimedia real-time applications, mobile stations should maintain a path to the Internet gateway with high quality while roaming around the network. We considered the path selection problem in IEEE 802.16 MMR networks and proposed a path selection metric that enables mobile stations to maintain high throughput and low delay.

7.2 Future Work

7.2.1 Real System Evaluation

Several research institutions have begun to build wireless mesh testbeds, such as Microsoft Mesh [25], MIT Roofnet [5], UCSB MeshNet [87], etc. Several companies [116, 117, 118, 27, 85] now provide mesh equipment and services to community and enterprise users. Software implementation and performance evaluation of our proposed frameworks in real word testbed and products are very interesting future work.

7.2.2 User-deployed Wireless Mesh Network

Generally speaking, there are two types of mesh network deployment. First, ISPs can deploy a mesh backbone for communities and enterprises. In this scenario, all the mesh routers are maintained and configured by the ISP and are beyond the control of normal users. Another type of deployment, known as user-deployed network, relies on

individual network users to contribute wireless routers and Internet uplinks. These independently owned and managed resources then self-organize to form a wireless backbone.

In this dissertation we have focused on ISP-deployed wireless mesh networks and proposed centralized algorithms. For user-deployed wireless mesh networks, it is more appropriate (also more challenging) to design distributed algorithms for rate allocation and QoS support.

7.2.3 Heterogeneous Wireless Networks

In order to provide stable and high-speed Internet access for mesh clients, inter-router and client-router communications usually use different radio technologies, which effectively eliminate interference between them. For example, IEEE 802.16 can be used for inter-router communication while client-router communication uses IEEE 802.11b. With the increased popularity of multi-interface devices and the software radio technology [119], future wireless devices can take advantages of multiple potential connections. For example, a mobile station can directly connect to the Internet gateway through WiMAX or connect to mesh routers through WLAN depending on the qualities of the connections. Handover and QoS maintenance are very interesting research topics for wireless mesh networks consisting of heterogeneous radio technologies.

BIBLIOGRAPHY

- [1] I. F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: a survey. *Computer Networks Journal*, March 2005.
- [2] Seattle wireless. <http://www.seattlewireless.net/>.
- [3] Champaign-Urbana community wireless project. <http://www.cuwireless.net/>.
- [4] Community wireless networking in western australia. <http://www.wafreenet.org/>.
- [5] MIT Roofnet. <http://pdos.csail.mit.edu/roofnet/doku.php>.
- [6] IEEE 802.11 Standard.
- [7] IEEE 802.16 Standard.
- [8] IEEE 802.16's Relay Task Group. <http://ieee802.org/16/relay/>.
- [9] R. Peterson, K. Johnsson, and entire Terminology Ad Hoc Group. Harmonized definitions and terminology for 802.16j mobile multihop relay. IEEE 802.16j-06/014r1, 2006.
- [10] S. Shenker. Fundamental design issues for the future internet. *IEEE JSAC*, 13(7):1176–1188, September 1995.
- [11] Kiyon, Inc. High quality voip over kiyon multi-service 802.11 wireless. White Paper, 2005.
- [12] V. Gambiroza, B. Sadeghi, and E. Knightly. End-to-end performance and fairness in multihop wireless backhaul networks. In *MobiCom*, 2004.
- [13] B. Wang and M. Mutka. QoS-aware fair rate allocation in wireless mesh networks. *Computer Communications (Special Issue: Resource Management and routing in Wireless Mesh Networks)*, 31(7), 2008.
- [14] B. Wang, M. Mutka, and E. Torng. Optimization based rate allocation and scheduling in TDMA based wireless mesh networks. In *ICNP*, 2008.
- [15] B. Wang and M. Mutka. Path selection for mobile stations in IEEE 802.16 multihop relay networks. In *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2008.

- [16] P. Kyasanur, X. Yang, and N. H. Vaidya. Mesh networking protocols to exploit physical layer capabilities. In *First IEEE Workshop on Wireless Mesh Networks*, Santa Clara, CA, September 2005.
- [17] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transaction on Information Theory*, 46(2):388–404, March 2000.
- [18] J. Li, C. Blake, D. S. J. D. Couto, H. I. Lee, and R. Morris. Capacity of ad hoc wireless networks. In *MobiCom*, 2001.
- [19] P. Bahl, R. Chandra, and J. Dunagan. SSCH: Slotted Seeded Channel Hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks. In *MobiCom*, Philadelphia, PA, September 2004.
- [20] P. Kyasanur and N. H. Vaidya. Capacity of multi-channel wireless networks: Impact of number of channels and interfaces. In *MobiCom*, Cologne, Germany, August 2005.
- [21] S. Yi, Y. Pei, and S. Kalyanaraman. On the capacity improvement of ad hoc wireless networks using directional antennas. In *MobiHoc*, Annapolis, Maryland, June 2003.
- [22] R. R. Choudhury and N. H. Vaidya. Performance of ad hoc routing using directional antennas. *Journal of Ad Hoc Networks*, November 2004.
- [23] J. A. Fuemmeler, N. H. Vaidya, and V. V. Veeravalli. Selecting transmit powers and carrier sense thresholds for CSMA protocols. Technical report, University of Illinois at Urbana-Champaign, October 2004.
- [24] X. Yang and N. H. Vaidya. On the physical carrier sense in wireless ad hoc networks. Technical report, University of Illinois at Urbana-Champaign, July 2004.
- [25] Microsoft Mesh Networks. <http://research.microsoft.com/mesh>.
- [26] D. B. Johnson, D. A. Maltz, and J. Broch. DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks. *Ad Hoc Networking*, pages 139–172, 2001.
- [27] Kiyon Autonomic Networks. <http://www.kiyon.com>.
- [28] C. E. Perkins and E. M. Royer. Ad hoc on-demand distance vector routing. In *2nd IEEE workshop on Mobile Computing Systems and Applications*, New Orleans, LA, February 1999.

- [29] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level measurements from an 802.11b mesh network. In *SIGCOMM*, Portland, Oregon, September 2004.
- [30] J. Bicket, D. Aguayo, S. Biswas, and R. Morris. Architecture and evaluation of an unplanned 802.11b mesh network. In *MobiCom*, Cologne, Germany, 2005.
- [31] R. Draves, J. Padhye, and B. Zill. Comparison of routing metrics for static multi-hop wireless networks. In *SIGCOMM*, Portland, OR, August 2004.
- [32] D. S. J. D. Couto, D. Aguayo, J. Bricket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *MobiCom*, San Diego, CA, September 2003.
- [33] S. Zhao, Z. Wu, A. Acharya, and D. Raychaudhuri. PARMA: A PHY/MAC aware routing metric for ad-hoc wireless networks with multi-rate radios. In *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2005.
- [34] B. Awerbuch, D. Holmer, and H. Rubens. High throughput route selection in multi-rate ad hoc wireless networks. In *Wireless On-demand Network Systems*, Madonna di Campiglio, Italy, January 2004.
- [35] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *MobiCom*, Philadelphia, PA, September 2004.
- [36] L. Qiu, P. Bahal, A. Rao, and L. Zhou. Troubleshooting wireless mesh networks. *SIGCOMM Computer Communication Review (CCR)*, October 2006.
- [37] P. Michiardi and R. Molva. Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Sixth IFIP conference on security communications, and multimedia*, Portoroz, Slovenia, 2002.
- [38] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Sustaining cooperation in multi-hop wireless networks. In *NSDI*, May 2005.
- [39] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *MobiCom*, 2000.
- [40] S. Buchegger and J-Y. Le Boudec. Performance analysis of the CONFIDANT protocol: Cooperation of nodes - fairness in dynamic ad-hoc networks. In *MobiHoc*, Lausanne, CH, June 2002.
- [41] S. Buchegger and J-Y. Le Boudec. A robust reputation system for P2P and mobile ad-hoc networks. In *Second Workshop on the Economics of Peer-to-Peer Systems*, June 2004.

- [42] B. Wang, S. Soltani, J. K. Shapiro, and P-N. Tan. Local detection of selfish routing behavior in ad hoc networks. In *International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN)*, Las Vegas, December 2005.
- [43] B. Wang, S. Soltani, and J. K. Shapiro. Local detection of selfish routing behavior in ad hoc networks. *Journal of Interconnection Networks*, 7(1), 2006.
- [44] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49:237–252, 1998.
- [45] S. H. Low and D. E. Lapsley. Optimization flow control — I: basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 7:861–874, 1999.
- [46] L. Chen, S. H. Low, and J. C. Doyle. Joint congestion control and media access control design for wireless ad hoc networks. In *INFOCOM*, 2005.
- [47] Z. Y. Fang and B. Bensaou. Fair bandwidth sharing algorithms based on game theory frameworks for wireless ad-hoc networks. In *INFOCOM*, 2004.
- [48] V. Gambiroza and E. Knightly. Congestion control in CSMA-based networks with inconsistent channel state. In *ACM International Wireless Internet Conference (WICON)*, Boston, MA, August 2006.
- [49] Y. Xue, B. Li, and K. Nahrstedt. Optimal resource allocation in wireless ad hoc networks: A price-based approach. *IEEE Transactions on Mobile Computing*, 2005.
- [50] B. Li. End-to-end fair bandwidth allocation in multi-hop wireless ad hoc networks. In *ICDCS*, 2005.
- [51] B. Radunovic and J. Y. L. Boudec. Rate performance objectives of multi-hop wireless networks. In *INFOCOM*, 2004.
- [52] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle. Layering as optimization decomposition: A mathematic theory of network architectures. *Proceedings of the IEEE*, 95(1):255–312, January 2007.
- [53] J.-W. Lee, M. Chiang, and A. R. Calderbank. Jointly optimal congestion and contention control based on network utility maximization. *IEEE Communication Letters*, 10(3), 2006.
- [54] G. S. Ahn, A. T. Campbell, A. Veres, and L. H. Sun. SWAN: Service differentiation in stateless wireless ad hoc networks. In *INFOCOM*, New York, June 2002.

- [55] Y. Yang and R. Kravets. Distributed QoS guarantees for realtime traffic in ad hoc networks. In *IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, 2004.
- [56] Y. Yang and R. Kravets. Throughput guarantees for multi-priority traffic in ad hoc networks. *Elsevier Ad Hoc Networks Journal*, 2006.
- [57] Y. Yang and R. Kravets. Achieving delay guarantees in ad hoc networks through dynamic contention window adaptation. In *INFOCOM*, 2006.
- [58] R. Nelson and L. Kleinrock. Spatial TDMA: A collision-free multihop channel access protocol. *IEEE Transaction on Communications*, 1985.
- [59] W. Wang, Y. Wang, X-Y. Li, W-Z. Song, and O. Frieder. Efficient interference-aware TDMA like scheduling for static wireless networks. In *MobiCom*, 2006.
- [60] D. Koutsonikolas, S. M. Das, and Y. C. Hu. An interference-aware fair scheduling for multicast in wireless mesh networks. *Journal of Parallel and Distributed Computing*, 2007.
- [61] C. T. Chou, A. Misra, and J. Qadir. Low latency broadcast in multi-rate wireless mesh networks. *IEEE Journal on Selected Areas in Communications (Special Issue on Multi-hop Wireless Mesh Networks)*, 24(11), November 2006.
- [62] P. Djukic and S. Valaee. Centralized scheduling algorithms for 802.16 mesh networks. In *Wimax/MobileFi: Advanced Research and Technology*. Auerbach Publications, CRC Press, 2007.
- [63] P. Djukic and S. Valaee. Link scheduling for minimum delay in spatial re-use TDMA. In *INFOCOM*, 2007.
- [64] H. Wei, S. Ganguly, A. Izmailov, and Z. Hass. Interference-aware IEEE 802.16 WiMAX mesh networks. In *61st IEEE VTC*, 2005.
- [65] D. Kim and A. Ganz. Fair and efficient multihop scheduling algorithm for IEEE 802.16 BWA systems. In *2nd International Conference on Broadband Networks*, 2005.
- [66] B. Han, W. Jia, and L. Lin. Performance evaluation of scheduling in IEEE 802.16 based wireless mesh networks. In *MASS*, 2006.
- [67] P. Djukic and S. Valaee. Delay aware link scheduling for multi-hop TDMA wireless networks. *IEEE/ACM Transactions on Networking*, 2009.

- [68] C. Molle, F. Peix, and H. Rivano. An optimization framework for the joint routing and scheduling in wireless mesh networks. In *19th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Cannes, France, September 2008.
- [69] V. Friderikos and K. Papadaki. Interference aware routing for minimum frame length schedules in wireless mesh networks. *EURASIP Journal on Wireless Communications and Networking*, 2008, 2008.
- [70] W. Wang, X. Liu, and D. Krishnaswamy. Robust routing and scheduling in wireless mesh networks. In *SECON*, 2007.
- [71] J. Zhang, H. Wu, Q. Zhang, and B. Li. Joint routing and scheduling in multi-radio multi-channel multi-hop wireless networks. In *2nd International Conference on Broadband Networks*, 2005.
- [72] M. E. Lubbecke and J. Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, November-December 2005.
- [73] Z. Walczak and J. M. Wojciechowski. Transmission scheduling in packet radio networks using graph coloring algorithm. In *International Conference on Wireless and Mobile Communications (ICWMC)*, 2006.
- [74] K. Saito and T. Inoue. Path selection in handover through RS. IEEE C802.16j-06/276, 2006.
- [75] K. Saito, T. Inoue, H. Kang, S. Lee, and H. Lim. Path selection for RS initial network entry. IEEE C802.16j-06/279, 2006.
- [76] H. Lee, H. Park, Y. Choi, Y. Chung, S. H. Rhee, Y. S. Lee, and Y. Kim. Link adaptive multi-hop path management for IEEE 802.16j. IEEE C802.16j-06/296.
- [77] H. Wang and P. Y. Kong. Data forwarding and routing path setup for IEEE 802.16j multihop relay networks. IEEE C802.16j-06/212r1, 2006.
- [78] V. Sreng, H. Yanikomeroglu, and D. D. Falconer. Relay selection strategies in cellular networks with peer-to-peer relaying. In *IEEE Vehicular Technology Conference*, 2003.
- [79] O. Oyman, S. Sandhu, N. Himayat, E. Visotsky, S. Ramachandran, and P. Sartori. End-to-end throughput metrics for QoS management in 802.16j MR systems. IEEE C802.16j-07/027r1, 2007.

- [80] I. D. Chakeres and E. M. Belding-Royer. Pac: Perceptive admission control for mobile wireless networks. In *1st International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine)*, Dallas, TX, October 2004.
- [81] Y. Yang and R. Kravets. Contention-aware admission control for ad hoc networks. *IEEE Transactions on Mobile Computing*, 4/4:363–377, 2005.
- [82] J. G. Augustson and J. Minker. An analysis of some graph theoretical cluster techniques. *Journal of the Association of Computing Machinery*, 17(4):571–586, 1970.
- [83] Reinhard Diestel. *Graph Theory*. Springer, 2005.
- [84] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [85] Tropos Networks. <http://www.troposnetworks.com/>.
- [86] Meshdynamics. <http://www.meshdynamics.com/>.
- [87] UCSB MeshNet. <http://moment.cs.ucsb.edu/meshnet/>.
- [88] R. Gupta and J. Walrand. Approximating maximal cliques in ad hoc networks. In *IEEE Personal, Indoor, and Mobile Radio Communications Conference (PIMRC)*, Barcelona, Spain, September 2004.
- [89] Z. Y. Fang and B. Bensaou. Design and implementation of a MAC scheme for wireless ad-hoc networks based on a cooperative game framework. In *IEEE Vehicular Technology Conference*, June 2004.
- [90] M. Fazel and M. Chiang. Network utility maximization with nonconcave utilities using sum-of-squares method. In *Conference on Decision and Control*, 2005.
- [91] D. Chiu and R. Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Computer Networks*, 1989.
- [92] <http://www.isi.edu/nsnam/ns/index.html>.
- [93] M. C. Domingo and D. Remondo. Analysis of VBR VoIP traffic for ad hoc connectivity with a fixed IP network. In *VTC*, 2004.
- [94] O. Malik. Google Launches WiFi Network in Mountain View. GigaOM. Available from <http://gigaom.com/2006/08/15/google-launches-wifi-network-in-mountain-view/>.
- [95] Google WiFi Mountain View. <https://wifi.google.com/support/>.

- [96] J. K. Shapiro, D. F. Towsley, and J. F. Kurose. Optimization-based congestion control for multicast communications. In *NETWORKING*, 2002.
- [97] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. In *MobiCom*, San Diego, CA, September 2003.
- [98] A. Bar-Noy, A. Mayer, B. Schieber, and M. Sudan. Guaranteeing fair service to persistent dependent tasks. *SIAM Journal on Computing*, 27:1168–1189, 1998.
- [99] D. E. Knuth. The sandwich theorem. *The Electronic Journal of Combinatorics*, 1:1–48, 1994.
- [100] F. M. Q. Pereira and J. Palsberg. Register allocation via coloring of chordal graphs. In *Asian Symposium on Programming Languages and Systems (APLAS)*, Tsukuba, Japan, November 2005.
- [101] M. Habib, R. McConnell, C. Paul, and L. Viennot. Lex-BFS and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing. *Theoretical Computer Science*, 234:59–84, 2000.
- [102] A. Kabbani, T. Salonidis, and E. W. Knightly. Distributed low-complexity maximum-throughput scheduling for wireless backhaul networks. In *INFOCOM*, 2007.
- [103] P. Heggernes. Minimal triangulation of graphs: A survey. *Discrete Mathematics*, 306:297–317, 2006.
- [104] D. Kratsch and J. Spinrad. Minimal fill in $O(n^2.69)$ time. *Discrete Mathematics*, 306(3):366–371, February 2006.
- [105] F. P. Kelly. Charging and rate control for elastic traffic. *European Trans. on Telecommunications*, 8:33–37, 1997.
- [106] F. Gavril. Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph. *SIAM Journal on Computing (SICOMP)*, 1(2):180–187, 1972.
- [107] NS2 simulator. <http://www.isi.edu/nsnam/ns/>.
- [108] C. Hoymann. Analysis and performance evaluation of the OFDM-based metropolitan area network IEEE 802.16. *Computer Networks*, 49:341–363, 2005.
- [109] J. G. Andrews, A. Ghosh, and R. Muhamed. *Fundamentals of WiMAX*. Prentice Hall, 2007.

- [110] Y. Ben-Shimol, I. Kitroser, and Y. Dinitz. Two-dimensional mapping for wireless OFDMA systems. *IEEE Transaction on Broadcasting*, 52(3), September 2006.
- [111] J. E. Wieselthier, C. M. Barnhart, and A. Ephremides. A neural network approach to routing without interference in multihop radio networks. *IEEE Transaction on Communication*, 42(1):166–177, 1994.
- [112] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2001.
- [113] S. Zhao, K. H. Teo, J. Z. Tao, J. Zhang, and T. Kuze. Macro diversity handover and fast access station switching for MMR network. IEEE C802.16j-07/119r1, 2007.
- [114] G. Q. Wang and al. MR Relay Link (R-Link) and Access Link monitoring and reporting procedure for Multi-hop path selectoin. IEEE C802.16j-07/213r4, 2007.
- [115] P. Mach and R. Bestak. Performance of IEEE 802.16 with relay stations. In *6th Conference on Telecommunications*, Peniche, Portugal, May 2007.
- [116] LocustWorld. <http://www.locustworld.com/>.
- [117] SkyPilot Networks. <http://www.skypilot.com/index.htm>.
- [118] Firetide. <http://www.fretide.com>.
- [119] M. Dillinger, K. Madani, and N. Alonistioti. *Software Defined Radio: Architecture, Systems and Functions*. Wiley, 2003.

MICHIGAN STATE UNIVERSITY LIBRARIES



3 1293 03062 6232