RESOURCE ALLOCATION IN WIMAX RELAY NETWORKS

by

Shen Wan

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

Doctor of Philosophy

in

Computer Science

MONTANA STATE UNIVERSITY
Bozeman, Montana

April, 2010

APPROVAL

of a dissertation submitted by

Shen Wan

This dissertation has been read by each member of the dissertation committee and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the Division of Graduate Education.

Dr. Jian (Neil) Tang

Approved for the Department of Computer Science

Dr. John Paxton

Approved for the Division of Graduate Education

Dr. Carl A. Fox

STATEMENT OF PERMISSION TO USE

DEDICATION

To my beautiful wife—Yihan Xiong,
my cute boy—Youti Wan,
and my forthcoming baby.

ACKNOWLEDGEMENTS

I would like to thank Dr. Jian (Neil) Tang for introducing me into this research project, for his technical insights and advices, for his help to improve my technical writing skills, for his personal example, kindness and support. He always encourages his students, leads by example, and puts students first. Thank you!

I would also like to thank Dr. Richard Wolff for teaching me fundamentals about wireless networks and bringing me onto wireless networking research, for his field expertise, supports, and suggestions.

I also thank: Dr. Yajing Xu, for all the cooperations in part of this research project; Dr. Brendan Mumey, for discussions, suggestions and technical validations; Li Zhang, for being a big help in and out of the lab; Chad Bohanan, for helps in the coding and presentation skills; and Seth Humphries for providing this wonderful LaTeX template.

I would also like to express much appreciation to my beautiful wife and my dear mom. They have given me countless help and support during this project.

Funding Acknowledgment

This research is supported in part by NSF grants CNS-0845776 and CNS-0519403, and a Montana State MBRCT grant #09-23. However, any opinions, findings, conclusions, or recommendations expressed herein are those of the author and does not reflect the position or the policy of the federal or state government.

TABLE OF CONTENTS

TABLE OF CONTENTS – CONTINUED

TABLE OF CONTENTS – CONTINUED

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

ABSTRACT

WiMAX is a promising wireless technology to provide high-speed, reliable communications in large areas. Relay stations can be deployed in a WiMAX network to extend its coverage and improve its capacity. Orthogonal Frequency Division Multiplexing Access (OFDMA) enables better channel utilization and exploits the benefits of channel diversity and user diversity. A smart adaptive antenna provides multiple Degrees of Freedom (DOFs) for intended communications and interference suppression, which results in more efficient spatial reuse and higher throughput.

In this dissertation, we discuss how to combine those aforementioned technologies in a WiMAX relay network (WRN), and study the routing, scheduling, channel assignment, and antenna DOF assignment problems.

It has been shown by previous research that the performance of wireless scheduling algorithms usually depends on the interference degree. Therefore, we study the interference degree in WiMAX networks and show that it is at most 4 in any 2-hop WRN and at most 14 in any general WiMAX network.

Next, we consider routing and scheduling in WRN with smart antennas. We formally define the Interference-aware Tree Construction Problem (ITCP) for routing, which offers full consideration for interference impact and DOF availability. We then present an algorithm to optimally solve it in polynomial time. As for scheduling, we first present a polynomial-time, optimal algorithm for a special case in which the number of DOFs in each node is large enough to suppress all potential secondary interference. An effective algorithm is then presented for the general case.

We also study a scheduling problem for throughput maximization in OFDMA-based WRN with consideration for multi-user diversity, channel diversity and spacial reuse. We present a Mixed Integer Linear Programming (MILP) formulation to provide optimum solutions. Furthermore, we show that both the simple greedy algorithm and our proposed weighted-degree based greedy algorithm have approximation ratio of $\frac{1}{5}$ for 2-hop WRN and $\frac{1}{15}$ for any WiMAX networks. In addition, we present 3 other efficient algorithms, namely, the maximum weighted independent set (MWIS) algorithm, the sequential knapsack algorithm and the LP rounding algorithm.

Extensive simulations are conducted to evaluate the performance of all proposed algorithms.

INTRODUCTION

WiMAX Background

The emerging WiMAX (Worldwide Interoperability for Microwave Access) technology (IEEE 802.16 [1]) is described by the WiMAX forum as "a standards-based technology enabling the delivery of last mile wireless broadband access as an alternative to cable and DSL." It can offer low-cost, high-speed and long-range communications for applications ranging from broadband Internet access to military and emergency communications.

A WiMAX network is composed of a Base Station (BS) and multiple Subscriber Stations (SS). The BS manages the WiMAX network and serves as a gateway connecting the WiMAX network to external networks such as the Internet.

Two operation modes are supported by the standard: the Point-to-Multipoint (PMP) mode and the mesh mode. Quite similar to a cellular network, a WiMAX network working in the PMP mode is essentially a single-hop wireless network in which an SS always directly communicates with the BS. In the mesh mode, a spanning tree rooted at the BS is formed for routing. An SS out of the transmission range of the BS can use other SSs as relay to communicate with the BS in a multihop fashion. Compared to the PMP mode, the mesh mode can significantly extend wireless coverage and improve network capacity.

WiMAX Relay Networks (WRN)

The IEEE 802.16j task group [2] has been formed to extend the scope of IEEE 802.16e [3] to support Mobile Multihop Relay (MMR) networks by introducing Relay

Stations (RSs) to WiMAX networks. A WRN is composed of a Base Station (BS), several Relay Stations (RSs) and many Subscriber Stations (SSs). A spanning tree rooted at the BS is formed for routing, in which all SSs are leaf nodes. If an SS is out of the transmission range of the BS, it can use one or multiple RSs to communicate with the BS in a multihop manner as illustrated in Fig. 1. Compared to a single-hop WiMAX network in which each SS directly communicates with the BS, a relay network can significantly extend the coverage range, reduce dead spots and improve network capacity [2].

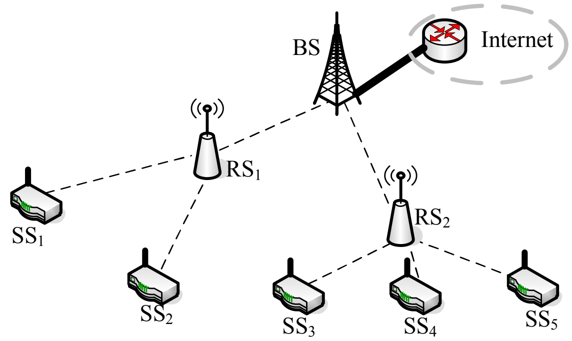


Figure 1: A WiMAX relay network

Unlike the WiFi(802.11) network, which uses Carrier Sense Multiple Access with Collision Detection (CSMA/CD) to reuse the wireless medium, the WiMAX standard [3] adopts a TDMA-based MAC layer in which the time domain is divided into minislots[1] and multiple minislots are grouped together to form a scheduling frame. The standard also provides a highly flexible approach that can be implemented over a wide range of frequencies and different physical layer technologies such as Orthogonal Frequency Division Multiplexing(OFDM), Multiple Input Multiple Output (MIMO) and smart antennas, some of which will be introduced in the following.

[1]Also known as “timeslots”

OFDMA

Orthogonal Frequency Division Multiplexing Access (OFDMA) is an emerging OFDM-based multiple access technology. Same as OFDM, the operating spectrum in OFDMA is divided into multiple narrow frequency bands (a.k.a. sub-carriers). However, unlike OFDM, which assigns all sub-carriers to a single user(link), OFDMA group sub-carriers into sub-channels, each of which consists of multiple sub-carriers. The multiple access is achieved by assigning different sub-channels to different users in the network for simultaneous transmissions. The available resources in an OFDMA-based WiMAX network can be viewed as *transmission blocks* (or simply *blocks*) in a 2-dimensional structure with minislots in one dimension and sub-channels in another [4], as illustrated in Fig. 2.

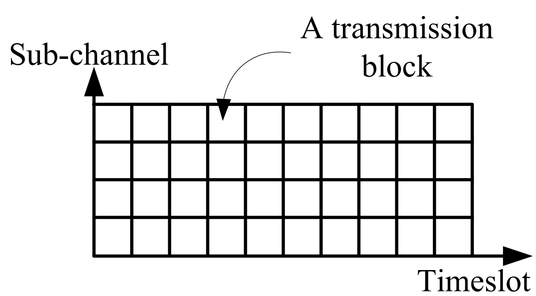


Figure 2: Transmission blocks

Hence, the scheduling problem in such a network is how to assign transmission blocks to each link in the network to optimize certain objective(s). We will discuss OFDMA channel model together with other channel models in detail in Chapter 3.

Smart Antenna

Unlike a conventional omni-directional antenna which wastes most of its energy in directions where no intended receiver exists, a smart (directional) antenna offers a longer transmission range and lower power consumption by forming one or multiple beams towards intended receiver(s) only. The emerging Digital Adaptive Array (DAA) antennas [5] can even perform fine-grained interference suppression by adaptively forming nulls in certain directions using its antenna elements[2], which leads to better spatial utilization. Therefore, smart antennas can enhance the functionalities of a WiMAX system and help it better achieve the goal of providing long-range and high-speed communications. The detailed antenna operation models will be discussed in Chapter 3. Although DAA antennas have been extensively studied before, research on relay networks using DAA antennas is still in its infancy.

Scope of this Dissertation

Although there are many challenges and important problems in WRN, we focus on resource allocation problems in this dissertation. We consider the 3 fundamental resources in a WiMAX network: the time, the spectrum, and the signal power.

How to manage the time and spectrum resource is a typical problem for signal multiplexing. Although WiMAX standard mandates to use TDMA for time resources and may apply OFDMA for spectrum resources, it does not address the details about how to assign minislots and sub-channels to different users. The standard leaves this problem open for device providers. Scheduling and channel assignment algorithms

[2] a.k.a, Degrees Of Freedom (DOFs)

need to be designed and implemented for the WiMAX protocol to work. And we will address this problem in this work.

How to utilize power to send signals and convey information is also essential for any wireless network. Traditionally, power control is considered as to decide how much power to use for transmitting signals at each transmitter. Many literature have been focused on this field. However, we do not cover this topic in this dissertation. Instead, we focus on how to better utilize the limited power resource, and send them only to desired receivers. In other words, we will study how to apply modern antenna technologies to improve the transmission efficiency and cancel possible interference, and in turn enhance the network performance. In this aspect, the basic resource to allocate is the flexibility to adjust antenna pattern at each smart antenna.

In a WRN, end-to-end paths need to be setup to realize any useful data flow. Therefore, we will also study the routing problem, which also decides the runtime topology of the network. For the routing problem, we consider only unicast traffic flows in one direction, either uplink to BS or downlink to SS, but not both of them simultaneously. This is due to the seperation of uplink subframe and downlink subframe in the WiMAX scheduling model. We also route all traffic flows at the same time in a centralized manner.

For each routing or scheduling problem, we will set up the system model, formalize the problem, examine its theoretical characteristic, propose exact, approximate, or heuristic algorithms to solve the problem, and justify the proposed algorithms with simulations. We do not use test beds or set up real networks to test our proposed algorithms.

Summary of Contributions

The WiMAX standard [1] specifies a common MAC protocol for both the PMP mode and the mesh mode, including signaling protocols and message structures. But the standard does not specify either the algorithm for computing transmission schedule (i.e., scheduling algorithm) or the routing protocol. The routing problem, i.e., the tree construction problem, has not been well studied for WRN, especially for those with DAA antennas. However, without careful consideration for interference impact and resource availability, scheduling transmissions along the constructed tree may lead to poor throughput and serious unfairness. In the simulation, we show that a Minimum Spanning Tree (MST) based routing approach performs very poorly.

Scheduling links with DAA antennas is quite different from traditional link scheduling with omni-directional antennas since simultaneous transmissions on two interfering links can be supported as long as DOFs are properly assigned to suppress interferences. The problem involves both link scheduling and DOF assignment, which makes it more challenging compared to scheduling with omni-directional antennas.

To our best knowledge, we are the first to address routing and scheduling problems in the context of WRN with DAA antennas. Our contributions are summarized in the following.

1. We discuss in detail about most popular system models used in the research field. For each model, we examine its realisticity and tractability and compare these attributes against those of counterpart models.

2. We show the importance of interference degree in wireless network scheduling problems and prove boundaries for interference degree in WiMAX networks.

This theoretical result can be used to evaluate the performance of many other heuristics and algorithms for WiMAX and other networks.

3. For routing, we formally define the Interference-aware Tree Construction Problem (ITCP), which offers full consideration for interference impacts and DOF availability. We present an algorithm to optimally solve it in polynomial time.

4. We consider the scheduling problem where the number of DOFs in each node is large enough to suppress all *potential secondary interference*. We present a polynomial-time optimal algorithm to solve it.

5. We present an effective heuristic algorithm for the general case of the joint scheduling and DOF assignment problem, and justify its performance by extensive simulations.

6. We present a Mixed Integer Linear Programming (MILP) formulation to provide optimum solutions for the joint scheduling and channel assignment problem, which can serve as benchmarks in performance evaluation.

7. We present many efficient heuristics, including two greedy algorithms, the Maximum Weighted Independent Set (MWIS) algorithm, the sequential knapsack-like algorithm, and the LP rounding algorithm, to solve the joint scheduling and channel assignment problem. Approximation ratios have been proved for the two greedy algorithms. We justify the efficiency of all algorithms by extensive simulation results. The idea of our LP rounding scheme is also applicable to similar ILP problems.

Outline of this Dissertation

The remainder of this dissertation is organized as following:

- We introduce related works in Chapter 2 and demonstate the novelty of this work to them.

- We discuss in detail about most popular system models used in the research field in Chapter 3. We compare the realisticity and tractability of each model and explain why we prefer some models to the others.

- In Chapter 4, we explain the concept of *interference degree* and why it is very important for wireless scheduling. Next, we summarize the existing research results on interference degree. Then we study the bound of interference degree in 2-hop WRN and general WiMAX networks respectively and prove our results.

- We consider how to construct a balanced routing tree with aware of interference for a WRN in Chapter 5. We propose a proved polynomial-time exact algorithm to solve this problem. The routing tree construction decides the topology of a WRN and provides essential inputs for the scheduling problems we consider later on.

- We consider joint scheduling and DOF assignment in Chapter 6. A polynomial-time exact algorithm is proposed for a special case of the problem. An efficient heuristic is proposed for the general case.

- We study the joint scheduling and channel assignment problem in Chapter 7. We formalize the problem in a Mixed Integer Linear Programming form and provide various approximate algorithms and heuristics to solve it in practice.

- In Chapter 8, we describe the simulation environment, tools and settings we used to evaluate all algorithms for our problems. Then we show extensive simulation results and analyze those results in detail. The simulation results justify the performance of our proposed algorithms.

- We summarize the conclusions of this disseration and discuss about some potential continuations of this research work in Chapter 9.

RELATED WORK

Related Work on WiMAX Scheduling

Transmission scheduling is a fundamental problem in OFDMA-based WiMAX networks, which has been studied by several recent works.

In [6], the authors presented one of the first transmission scheduling algorithms in single-channel WiMAX mesh networks, which has been shown to achieve high channel efficiency and provide fair access to all nodes in the network by simulation results.

In [7], the authors focused on Quality of Service (QoS) support and proposed routing and scheduling algorithms to provide per-flow QoS guarantees.

In [8], Cao *et al.* introduced a new fairness notion and imposed it contingently on actual traffic demands. They presented an optimal algorithm to solve a scheduling problem with objective of maximizing network throughput within their fairness model.

In [9], a distributed algorithm was presented to provide fair end-to-end bandwidth allocation for single-radio, multi-channel WRN.

In [4], Andrews and Zhang considered several scheduling problems in single-hop OFDMA-based WiMAX networks. They analyzed the hardness of these problems and present several simple constant factor approximation algorithms to solve them. Furthermore, they considered the problem of creating template-based schedules for such networks in [10]. They presented a general framework to study the delay performance of a multi-carrier template. They then described some deterministic and randomized scheduling algorithms for template creation and studied their delay performance via analysis and simulation.

The authors surveyed various research efforts on WiMAX scheduling in [11]. Both the mesh network and relay network is mentioned in the paper and OFDM/OFDMA channel is discussed.

Related Work on Relay Networks

The general link scheduling for multihop wireless networks with omni-directional antennas has been studied in [12, 13].

Transmission scheduling has also been studied for single-carrier or OFDM-based WiMAX multihop networks. Various centralized heuristics have been proposed for scheduling and/or routing in [14, 6, 15, 16] with the objective of maximizing spatial reuse.

In [17], Sundaresan *et al.* showed that the scheduling problem to exploit diversity gains alone in a 2-hop 802.16j-based WRN is NP-hard. They then provided polynomial-time approximation algorithms. They also proposed a heuristic algorithm to exploit both spatial reuse and diversity.

In [18], the authors studied a similar scheduling problem in OFDMA-based relay networks. They provided an easy-to-compute upper bound on the optimum. They also presented three fast heuristics for the problem, and showed by simulation results that those heuristics perform close to the optimum and outperform other existing algorithms.

In [19], the authors studied the scheduling problem in 2-hop OFDMA relay networks and proposed approximation algorithms to optimize the end-to-end throughput within a propotional fairness model.

In [20], the authors studied joint allocating data rates and finding a stabilizing scheduling policy in a multihop wireless network. A dual optimization based approach was proposed to fully utilize the network capacity and maintain fairness.

In [21], joint scheduling and routing in relay-assisted broadband cellular network was studied. A centralized opportunistic algorithm was proposed to solve it.

Related Work on Smart Antennas

Several recent works [8, 9, 14, 6, 7, 15, 16] have addressed the problem of scheduling transmissions along a routing tree for WRN with omni-directional antennas. However, smart antennas have also received tremendous attention due to their capabilities of range extension, power saving and interference suppression.

MAC protocols were proposed in [22, 23] for 802.11-based ad-hoc networks using either switched beam antennas or adaptive antennas. The authors modified the original 802.11 MAC protocol to exploit the benefits of smart antennas.

In a recent work [5], the authors considered the problem of determining DOF assignment for DAAs with the objective of interference minimization. They presented constant factor approximation algorithms to solve it. Moreover, they proposed a distributed algorithm for joint DOF assignment and scheduling.

In [24], we studied routing and scheduling in WRN with smart antennas.

Another important type of smart antenna is the Multiple Input Multiple Output (MIMO) antenna, which supports multiple concurrent streams over a single link.

In [25], the authors discussed key optimization considerations, such as spatial multiplexing, for MAC layer design in ad-hoc networks with MIMO links. They presented a centralized algorithm and a distributed protocol for stream control and medium access with those key optimization considerations incorporated.

A constant factor approximation algorithm was proposed for a similar problem in [26].

A unified representation of the physical layer capabilities of different types of smart antennas was presented in [27] with some unified medium access algorithms.

In [28], Hu and Zhang examined the impact of spatial diversity on the MAC design, and devised a MIMO MAC protocol accordingly. They also studied the impact of MIMO MAC on routing and characterized the optimal hop distance that minimizes the end-to-end delay in a large network.

Cross-layer optimization for MIMO-based wireless networks has also been studied in [29, 30]. In [29], Bhatia and Li presented a centralized algorithm to solve the joint routing, scheduling and stream control problem subject to fairness constraints.

Novelty of this Work

The differences between our work and these related works are summarized as follows:

1. As mentioned before, due to the interference suppression feature of DAA antennas, the joint scheduling and DOF assignment problem studied here is significantly different from the scheduling problems with omni-directional or MIMO based antennas.

2. Our optimization goal is to improve end-to-end throughput and fairness. However, the general scheduling problems studied in [31, 23, 26, 25, 27, 5] aim at maximizing single-hop throughput or minimizing the frame length.

3. The routing tree and transmission schedule computed by our algorithms have certain good, provable properties which cannot be supported by the heuristic algorithms reported in [9, 14, 6, 7, 15, 16].

4. As pointed out by [4, 18, 17], the joint scheduling and channel assignment problem considered in this work is different from the scheduling problems in single-carrier or OFDM-based WRN [8, 9, 14, 6, 7, 15, 16], in which every link works on the same channel or can only be assigned a single channel in a minislot.

5. Spatial reuse is one of the main concerns of this work, which was not addressed by [4, 10, 18].

6. The closely related work [17, 19] only considered scheduling and channel assignment in a 2-hop WRN. In this work, we address a general WiMAX network without hops limit.

SYSTEM MODELS

Physical Layer Models

Radio Models

Radio, or transceiver, is the fundamental device for wireless communication. There are varient definitions for a radio, but in this work, we define it as the electronic device that transmits/receives signals to/from the wireless medium through one or multiple antennas, and transforms signals between wired circuits and wireless medium. Radios can be categorized in many ways, however, we focus on the relationship of radios to communication nodes. In this aspect, there are two mainstream radio models.

Half-duplex Radios: This model assumes that each node is equipped with a half-duplex radio, which can be tuned in either transmitting mode or receiving mode. However, the radio cannot transmit and receive simultaneously, not even with different channels for transmitting and receiving. Therefore, in a network with such half-duplex radio contraint, all the incoming links of a node are essentially conflicting with any of the outgoing links of the same node, and cannot be activated at the same time. On the other hand, it is still possible to have multiple incoming (or outgoing) links activated at the same time, if they are working on othorgonal channels. Thus, for RSs with both incoming and outgoing links, such asymmetry will make the scheduling problem more difficult for a multi-channel system.

The modern electronics technology allows the switching between transmitting mode and receiving mode to take less than 100 μs, which is negligible compared with the frame duration of WiMAX system at about 5–10 ms[1]. Half-duplex radios are easier to manufacture and cost less. Thus it is reasonable to assume that MSs are equipped with them.

Full-duplex Radios: As opposed to half-duplex radios, full-duplex radios support transmitting and receiving signals simultaneously. If all RSs are equipped with full-duplex radios, the incoming links and outgoing links can be scheduled indiscriminately. Two links can be scheduled at the same time if they are assigned different channels, no matter whether they are incoming or outgoing links. Full-duplex radios are more complex than half dulplex radios and therefore more expensive. However, it is still reasonable for RSs to be equipped with full-duplex radios because they are part of the infrastructure and are not as cost sensitive as the end terminals (MSs). It is also possible for an RS to have two half-duplex radios working coordinatedly, instead of having a full-duplex radio. For the purpose of this work, we consider these two approaches the same in our system model.

It is worth pointing out that a network can only benefit from full-duplex radios if it is a multi-channel system. In a single-channel system, the transmission of a node will severely interfere with signals arriving at the same node. Even if the node has two half-duplex radios instead of one full-duplex radio, the interference will still be much stronger than the desired signal, since the distance between two radios is much shorter than the distance between different nodes in the network. For such a close distance, even directional antenna are unlikely able to cancel the interference.

Our Radio Model: Although two half-duplex radios can emulate one full-duplex radio in a node, there is almost no benefit of having more than two radios in a node. In order to simultaneously transmit to and receive from multiple nodes, a node only needs to have one full-duplex radio with multiple directional antennas or a smart antenna that is capable of forming multiple beams. We will elaborate about antenna models in the following.

In summary, we assume in our system model that BS and MSs in a WRN are equipped with one half-duplex radio for each node. Whereas RSs are equipped with full-duplex radios in a multi-channel system, and half-duplex radios in a single-channel system.

Antenna Models

The antenna is an essential part of a wireless node. All signals are transmitted to or received from wireless medium via antennas. The profile of antenna is one of the most important factor to decide how much signal gain can be achieved in transmissions and how much the transmission will be affected by interference sources. Recently, many new antenna technologies have evolved and been applied to various wireless networks. WRN can also adopt those technologies to provide better services. We will discuss several popular antenna technologies and corresponding models in the following.

In reality, the pattern of any antenna is a function of antenna gain to the angle in a 3-dimensional space. However, most research in wireless networking considers that all nodes within the interference range are roughly in a plane. This is a reasonable assumption since the height difference of nodes usually spreads from a couple meters to tens of meters, whereas the communication/interference range is usually at least several hundreds of meters. With such assumption, the antenna pattern considered in wireless networking is simplified to a function of antenna gain to the angle in a 2-dimensional space. Different antenna models assume different characteristics of the antenna pattern function.

Omni-directional Antennas: These antennas radiate power uniformly in a plane and therefore can be modeled easily. The antenna gain is homogeneous in all directions, which indicates that the received signal or interference power is only related to the positions of transmitter and receiver. Omni-directional antennas are also the simplest and cheapest type of antennas, and still the dominant type of antennas in WiMAX and many other wireless networks.

In addition to omni-directional antennas, there are also sector antennas which emit power almost uniformly in a sector-shaped region. They are widely used in BSs of cellular systems, in which all interested communication nodes are located within a sector, or there are multiple sector antennas equipped on a BS and they can cover all directions coordinately. Although using sector antennas may impact the network topology design, they do not affect the scheduling or routing process within a sector. We do not consider scheculing cross multiple sectors. Therefore, we consider sector antennas the same as omni-directional antennas.

Omni-directional antenna does not affect scheduling and therefore is the simplest model. Since it is still the most popular antenna technology, omni-directional antenna model is also a realistic antenna model and the most popular model in wireless network research, including the WiMAX field.

Directional Antennas with Fixed Pattern: A directional antenna can emit radiation only at directions toward intended receivers. Thus it can utilize the transmission power more efficiently, achieve a longer transmission range, improve the signal to noise ratio, reduce power consumption, and even suppress interferences. The antenna pattern and the antenna gain with regard to direction are related. Although the antenna pattern is in reality a continuous function over the angle (direction), people usually model it as several beams and nulls, each having a direction, an angle spread and a gain value. At the direction a beam pointing to, the antenna gain is large, such that signals from this direction can be strengthened. Of course, interference from the same direction will also be strengthened, and therefore needs other means to resolve. With fixed antenna pattern, the angle spread (beam width) and the gain of each beam or null is predetermined by hardware. The relative angles between each pair of antenna beams or nulls are also fixed. However, the antenna may be able to "rotate" the antenna pattern on the plane, either mechanically or electronically. A directional antenna is supposed to point its beams to the communication targets, and may use its nulls to cancel interferences coming from undesired directions. Since the desired communication counterparts and undesired interference sources change from minislot to minislot, the antenna pattern "rotation" is interdependent with the scheduling problem. Therefore, this model is much harder to schedule with than the omni-directional antenna model. This model also has the disadvantage that a fixed antenna pattern cannot support an arbitrary network topology efficiently, since the angles of neighboring nodes to a node are also arbitrary, and may not fit the angles between the fixed beams/nulls.

<u>Directional Antennas with Adjustable Pattern, Fixed Beamwidth</u>: In this model, the constraint of fixed antenna pattern is relaxed, such that an antenna can point its beams and nulls toward any direction. In practice, the angle between a beam and a null cannot be arbitrarily small. However, this constraint is usually not considered in scheduling research. This is because firstly, in a real network topology, a very small

angle between a beam and a null is rarely necessary [1] ; and secondly, it would make the joint schedule and antenna pattern forming problem too complicated.

This model is quite practical because it closely models a type of smart antenna—the Digital Adaptive Array (DAA) antenna. A DAA antenna has multiple antenna elements, each of which can emit a signal independently. By carefully controling the amplitude and phase of the electro-magnetic waves generated by each element, the DAA antenna in a whole can form multiple independent beams or nulls. A DAA antenna with K elements is said to have K Degrees of Freedom (DOFs) because the number of beams and nulls it can form equals to K.

A common DAA antenna approach is to use one DOF to form a beam toward intended communication counterpart, such that the Signal-to-Noise-Ratio (SNR) can be improved, and therefore the transmission range can be increased. In order to enable communication between a pair of transmitter and receiver (v_i, v_j), a single DOF needs to be assigned for communications at each end. Moreover, except for the DOF assigned for communication, the remaining $(K - 1)$ DOFs can be used at the transmitter to cancel its interference to other receivers by forming nulls at corresponding directions. Similarly, the receiver can use its remaining $(K - 1)$ DOFs to suppress interference from other transmitters. At directions where neither beam of null has been assigned, communication is impossible but there may still exist interference.

It is worth pointing out that it is both impossible and useless to use all DOFs to form nulls without forming one beam. On the other hand, forming multiple beams

[1] Consider a node u with a beam pointing toward another node v of its neighbor, u needs to form a null toward some node w only if u and w are withing interference range and w is neither u's nor v's neighbor, otherwise there exist collision which forming null cannot help resolve. This situation is very unlikely to happen in a tree topology.

may only be desirable for multi-cast communications, or uni-cast in a multi-channel system. This is because, with only one channel, it is impossible to transmit to several neighbors simultaneously.

Although a DAA antenna can generate beams and nulls of different beam widths, people usually want the beam width to be as narrow as possible, since that will give a largest antenna gain for the beams and smallest antenna gain for the nulls. However, due to the limitations of physics law and electronics technology, the beam width cannot be arbitrarily narrow. Thus, the antenna pattern model of a DAA antenna can be considered to have fixed antenna beam/null width.

In such a model, the antenna pattern forming is related to the communication links needed to activate and the interference canceling, which are in turn related to the scheduling. Therefore, this model is also much harder to schedule with than the omni-directional model.

Directional Antennas with Adjustable Pattern and Beamwidth: It is also possible to assume a beamwidth adjustable antenna pattern, such that the antenna may be able to use a beam to cover multiple communication counterparts or use a null to cancel interference from nodes in close directions. The network capacity can be further exploited with such antenna model. However, this model is too hard to solve because different beam width will result in different communication/interference ranges, and in turn makes antenna pattern forming coupled with both scheduling and topology control, which results in a extremly difficult problem.

Our Antenna Models: We want to use antenna models that are both practical and mathematically tractable. Therefore, we consier two antenna models: the omni-directional antenna and the directional antenna with adjustable antenna pattern and fixed beamwidth. In real WRN, the BS may be equipped with an omni-directional antenna or multiple sector antennas. The RSs may be equipped with a DAA antenna and change its antenna pattern in minislots. Although SSs may also benefit from DAA antennas, they may be equipped with only omni-directional antenna for price concerns.

Channel Models

A channel model desides how to use the available spectrum. The spectrum can either be used as a whole, which forms a single channel; or divided into multiple channels, and thus enabling the Frequency Duplex Multiple Access (FDMA). The channels divided from a spectrum are usually assumed to be orthogonal to each other, which means transmissions in different channels can be considered interference-free.

Single Channel: For a system with only one channel, there is no channel assignment problem. Whether two links are possible to interfere with each other only depends on their relative positions. When two links are possibly interfering, however, antenna nulls may be used to cancel the interference. We will discuss this technique in the following chapters. The single channel model is the simplest to study and forms the fundation of all other channel models. It is also a practical model since many real networks actually have just one channel, or only one in a scheduling period.

OFDM Channels: The Orthogonal Frequency-Division Multiplexing (OFDM) technology utilizes a large number of closely-spaced orthogonal sub-carriers[2] to transmit data. Each sub-carrier uses a conventional low-datarate modulation scheme, but the large number of sub-carriers compensates for the low rate. The OFDM technology has many advantages over the single wide channel solution, such as robustness against channel interference and inter-symbol interference, high spectrum efficiency,

[2] A sub-carrier can be imagined as a narrow-band channel.

and more efficient electronics implementation. However, all OFDM sub-carriers are used to serve a single user. Thus, it is similar to a single channel system from the perspective of scheduling. In this work, we do not distinguish the OFDM channel model from the single channel model.

Homogeneous OFDMA Channels: Similar to the OFDM technology, Orthogonal Frequency-Division Multiple Access (OFDMA) utilizes lots of orthogonal sub-carriers. However, instead of assigning all sub-carriers to serve a single user as in OFDM, OFDMA groups sub-carriers into several non-intersecting subsets, and assigns one subset of sub-carriers to a user. In reality, sub-carriers are often grouped into sub-channels[3] first, then users are assigned to different sub-channels.

In a homogeneous OFDMA channel model, all sub-channels are considered able to achieve the same data rate for each user. Therefore, the datarate achievable on any user at a minislot only depends on the number of sub-channels assigned to that user. In this sense, all sub-channels are homogeneous and the joint scheduling and channel assignment problem is essentially just a simple one-resource scheduling problem because sub-channels in a minislot can be considered as multiple smaller minislots. Therefore, although with homogeneous OFDMA channels, channel assignment must be done in addition to the time scheduling, it does not actually generate a more difficult problem.

Heterogeneous OFDMA Channels: The assumption that all sub-channels can provide same datarate to a user simplifies the scheduling. However, in reality interference level may be different on various sub-carriers or sub-channels. Multipath fading and user mobility lead to independent fading across users for a given sub-channel. Therefore, the gains of a given channel for different users vary, which is referred to

[3]Since sub-carriers are orthogonal to each other, so are sub-channels.

as *multi-user diversity* [17]. Moreover, different channels may experience different channel gains for a given SS due to the frequency selective fading, which is referred to as *channel diversity* [17]. In addition, multiple users that are far apart can reuse multiple sub-channels simultaneously, which is referred to as *spacial reuse*. In order to design an efficient scheduling algorithm for an OFDMA-based WRN, we need to exploit the benefits that can be provided by multi-user diversity, channel diversity and spatial reuse. Although it is possible to use per sub-channel power adaptation to further exploit channel diversity, we assume in this work that the transmit power is fixed and split equally across all sub-channels.

With these factors considered, the achievable data rates for a user can actually be quite different for different sub-channels. With such heterogeneous OFDMA channel model, one of the most important question is how to assign a sub-channel to the most appropriate user, such that it is most efficient for the entire network. This is usually not an easy job if interference is considered, as channel assignment is coupled with scheduling and spatial reuse. We will show the hardness of such a problem in the following of this work.

Transmission Models

The transmission model decides whether a node can successfully transmits data to another node on some channel(s). We only consider the Signal-to-Noise Ratio (SNR) instead of the more general Signal-to-Interference-and-Noise Ratio (SINR). Therefore, we can formalise the transmission model as a directional graph $G = (V, E)$, whereas each node corresponds to a vertex $v_i \in V$, and there exists a directional link e from node v_i to v_j if and only if v_i is able to transmit data to v_j. All such directional links correspond to the edge set E in G. This graph is usually called a *communication graph*. Note that the communication graph is directional in general, because power level may be different at the transmiters side and noise level may also be different at the receivers side.

Unit Disk Graph: If all nodes are placed on a plane with no blockage in between and transmit using the same fixed power, then each node will have a uniform transmission range R_T. In this case, $(v_i, v_j) \in E$ if $\|v_i - v_j\| \leq R_T$.[4] This is the simplest transmission mode and is usually called the unit disk graph model because it is effectively the same as placing a unit-sized (radius $r = R_T$) disk centered at each node, and adding a link between two nodes if and only if the disk belonging to one node can cover the other node. Note that in unit disk graph model, the resulting communication graph is undirectional. With such a model, many geometric theorems may be used to deduce the characteristic of the underlying network, and those characteristics can decide the network capacity, or the efficiency of some scheduling algorithms, as shown in the following part of this work as well as in many other research works. The major assumptions made by this unit disk graph model are that 1) power control does not exist and 2), the wireless medium is considered homogeneous in every direction, which is true in free space but may not be precise enough if terrain effects are considered. The unit disk graph model is widely used in the wireless networking research field because it is easy to analyze while still close enough to the reality.

Geometric Intersection Graph: If nodes can transmit with different power levels, then the unit disk graph becomes graph generated by disks with various sizes. In this geometric intersection graph model, geometric attributes may still be used to deduce the characteristics of the underlying network. However, it is much harder than the deduction with the unit disk graph model. Note that unlike in the unit

[4] $\|v_i - v_j\|$ indicates the Euclidean distance between node v_i and v_j.

disk graph model, the communication graph may be directional in the geometric intersection graph model, since nodes with higher power may be able to reach nodes with lower power level, but not vice a versa. A simplification of this model would be to assume only several fixed levels of transmitting power. A reasonable assumption is that BSs and RSs can have higher transmission power whereas MSs can only afford lower transmitting power.

Arbitrary Graph: Usually, we can assume that wireless signal transmission in the atmosphere is close enough to the transmisson in free space. Even if the attenuation exponent may be higher than that of 2 in free space, it is still considered a constant in all directions. However, when the network is deployed in an area with rugged terrain, it may not be possible to have Line-of-Sight (LOS) transmission in all directions. When LOS is not available, either transmission becomes impossible or signals must go through other paths with reflection and therefore have lower SNR and in turn shorter communication range. Furthermore, multi-path and terrain effects can also be significant to signal propagation, especially if the terrain includes large area of water surface. With these factors considered, the effective communication range will be less than or equal to that in free space, and therefore the resulting communication graph will be a subgraph of that in unit disk graph model.

If the terran factors are considered together with variable power level, then the communication graph can be an arbitrary directional graph, which is the most difficult transmission model for a scheduling problem.

Interference Models

No Spatial Reuse: This model does not utilize any spatial reuse. In any mini-slot, there can be at most one link active on any channel within the entire network.

Some trivial scheduling algorithm in the WiMAX standard[1] adopts this model. Of course, schedules computed by algorithms following this model will always be feasible. In some networks, such as a PMP WiMAX network, or some 2-hop WRN that demands high SINR, this model fits very well. However, in most other cases the model is too pessimistic, such that a lot of spectrum resourses are wasted. Except for benchmark purposes, the performance of algorithms following this model is of little interest.

Primary Interference: This interference model actually does not consider general interference. Instead, it simply treats collisions as the only interference source. If two links $e = (v_i, v_j)$ and $e' = (v_{i'}, v_{j'})$ are incident, we say there exists *primary interference* in between. In this case, in no way can these links be active concurrently due to the half-duplex (a transceiver can only transmit or receive at one time), unicast (a transmission only involves a single intended receiver) and collision-free (two transmissions intended for the same receiver cannot happen at the same time) constraints. In Fig. 3, all the blue dash-dotted lines are those links that have primary interference with the link (T, R) shown in red solid line.

Although this model is obviously too optimistic, it does have some value. First, it is usually much easier to design scheduling algorithms and analyze the network performance with such a model, because scheduling problems can often be formalized as maximum matching problems instead of maximum independent set problems. Second, the performance achieved with such model can serve as an upper limit and benchmark for other algorithms with more practical models. And finally, with smart antenna technology, it is possible to cancel all potential interference by utilizing antenna nulls, in which case only primary interference will hamper the spatial reuse. Researchers used primary interference model in [32] and achieved algorithms with

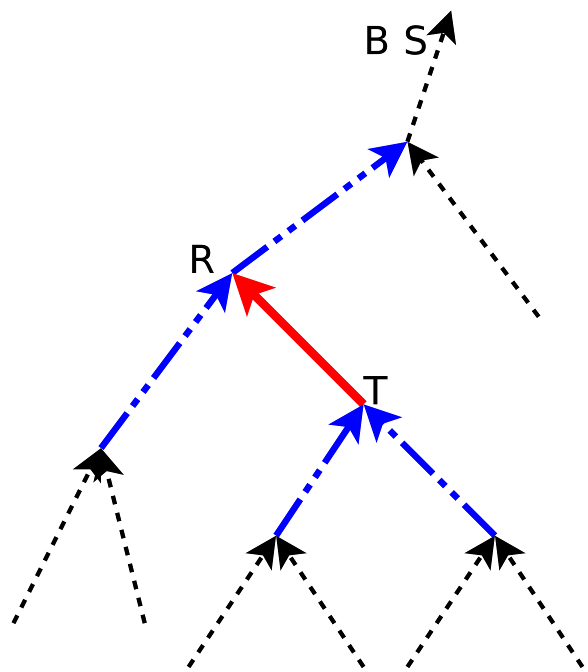


Figure 3: Primary interference versus 1-hop interference

constant approximation ratios and low time complexity. We will discuss problems with such an assumption more intensively in the following of this work.

1-hop Interference: Since the primary interference model is too optimistic, people have proposed 1-hop interference model, which, as its name suggests, assumes that 2 links interfere with each other if and only if they are at most 1-hop away from each other. In Fig. 3, all the other links have 1-hop interference with the link (T, R) shown in red solid line.

The 1-hop interference model is considerably more difficult than the primary interference model. However, there are still some graph theory techniques that can be used to help solving the problem or reaching some approximation algorithms. The 1-hop interference model is suitable for networks with an RTS/CTS (Request-To-Send/Clear-To-Send) based MAC layer. In such a MAC protocol, data transmission needs a 3-way hand-shaking process, which is vulnerable to collision if any node of one link is within communicaion range of some node of the other link. In [33], the authors used an equivalent 1-hop interference model to study the MAC layer capacity of ad hoc wireless networks. The problem was reduced to a maximum distance-2 graph matching problem and solved with approximation algorithms. However, since WiMAX adopts a TDMA-based MAC layer, 1-hop interference model is not appropriate.

<u>k-hop Interference</u>: To generalize from the 1-hop interference model, k-hop interference models are proposed. Sometimes these models are used to approximate the interference range model [34], where $k \equiv \left\lceil \frac{R_I}{R_T} \right\rceil$. Researchers proved in [35] that a maximum throughput scheduling problem in wireless networks with such k-hop interference can be reduced to a maximum weighted matching problem, which is NP-hard and cannot even be approximated within a factor that grows polynomially with the number of nodes. Distributed wireless scheduling algorithms were proposed in [34] based on the k-hop interference model.

These models are even harder than the 1-hop interference model. It is usually very hard to achieve exact algorithms or even approximate algorithms with reasonable approximation ratios for a problem with such an interference model. Despite the hardness, k-hop interference models are not as realistic as the interference range model, and are therefore rarely used in WiMAX research.

<u>Interference Range Model</u>: Instead of using the number of hops between two links as the metric, the interference range model decides whether interference may exist by the physical distance between end nodes of these two links. In this model, a network-wise constant interference range R_I is specified, which is normally 2–3 times larger than the communication range R_T. For two unincident directional links

$e = (v_i, v_j)$ and $e' = (v_{i'}, v_{j'})$, they may interfere with each other if $\|v_i - v_{j'}\| \le R_I$ or $\|v_j - v_{i'}\| \le R_I$. We say that there exists *secondary interference* between e and e' in such case. For example, consider all links interfering with the link in red solid line in Fig.4. The two circles in dashed lines center at the transmitter and receiver of the red link, respectively. The circles have radius R_I. All the blue dash-dotted links have primary interference with the red link. And all the black solid links have secondary interference with the red link. Note that there are two green dashed links which do not have any interference with the red link under this interference model.

If, instead, all links are considered undirectional, then there also exists secondary interference if $\|v_i - v_{i'}\| \le R_I$ or $\|v_j - v_{j'}\| \le R_I$. In the above example, those two green dashed links would then have secondary interference with the red link.

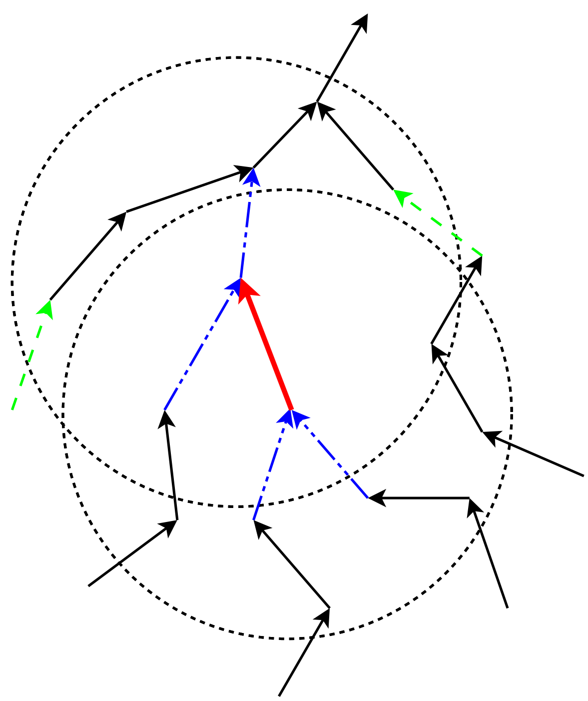


Figure 4: Illustration for secondary interference

In WiMAX, TDMA-based MAC is adopted and the scheduling frame is split into a downlink subframe and an uplink subframe. Therefore links are considered directional. Note that even if two links have secondary interference in between, they may still be able to transmit data simultaneously if different channels are assigned to them or smart antenna technology is used to cancel interference. With a DAA antenna, a DOF can be assigned at either the transmitter v_i or the receiver $v_{j'}$ to suppress secondary interference from e to e'. Similarly, a DOF can be assigned at either the transmitter $v_{i'}$ or the receiver v_j to suppress secondary interference from e' to e.

The interference range model is closer to reality and also more difficult. Graph theory can hardly be applied to such model because two links that are arbitrarily far away from each other in terms of hop count may still be within the interference range. Without graph theory's help, only some geometric theorems may be used to attack problems with this model. We will show how geometry is used to achieve some theoretical results under such an interference model.

The interference range model is widely used in WiMAX research. The major disadvantage is that, due to its hardness, it is very difficult to design algorithms with any theoretical merit. Usually researchers will only propose heuristics and analyze their performance through numerical analysis or simulations. In this work, however, we achieved some theoretical results on the approximation ratios of greedy-based algorithms for the joint scheduling and channel assignment problem, which will be introduced in the following.

<u>Protocol Interference Model</u>: If we put power control and terran effects into consideration, then the interference range in a network will not be constant. In order to decide whether two links will interfere with each other, the Signal-to-Interference-

and-Noise Ratio (SINR) needs to be computed at each receiving node for all pairs of links. The results can be modeled as an interference graph, in which each vertex corresponds to a communication link, and there exists an edge between two vertices if and only if the corresponding links interfere with each other. Equivalently, we can identify for each node v_i a set N_i of neighboring nodes potentially interfering with it, using the method introduced in [5]. Briefly, $v_j \in N_i$ if the SINR at receiver v_j (or v_i) will drop below the threshold due to interference from v_i (or v_j). As we will show later, the scheduling problem with such an interference model can often be formalized as maximum independent set problems on the interference graph, which is well known to be NP-complete. This model is very general; actually all models mentioned before are special cases of this model. Due to its hardness, the protocol interference model is rarely used in WiMAX research.

Physical Interference Model: Although the protocol interference model is very general and realistic, it only considers pair-wise interference among all links. In reality, a link will receive interference signals from all simultaneously active links on the same channel. And those signals all contribute to the interference part in the SINR computation. Therefore, the physical interference model considers a set of links. They can be active at the same time only if the SINR at every receiver of those links can remain above a threshold. This model is extremely hard for any network with reasonable size because it is even NP-complete to enumerate all such realisable transmission link sets. The application of smart antennas will make the problem even harder as a receiver may only need to nullify some of its interference sources. To the best of our knowledge, the physical interference model has only been used in simulations. No algorithm has ever been designed catering to this over-complicated model.

Summary

The physical layer is the lowest and most fundamental layer in a network model. It is also the most complicated layer. We have discussed many issues at the physical layer, including the radios, smart antennas, channels (sub-channels/sub-carriers), signal transmission and propagation, terrain effects, interferences, etc. And there are even more physical layer factors/technologies that may affect the WiMAX system design and performance: power control, Multiple-Input Multiple-Output (MIMO) antennas, compress-and-forward relaying, and modulations to name a few. It is simply impossible to take everything into consideration, so we just leave these issues out of the scope of this work. We assume that the concerned attributes of a link, including communication range, data rate, delay, etc., can be fully decided if all the physical layer models and their corresponding parameters are given. The objective of higher layers is to decide how to utilize the communication capacity provided by the physical layer, and how to allocate these resources to different users/traffic flows.

Data Link Layer Models

Media Access Control (MAC) Sublayer Model

The WiMAX standard [1] adopts a Time Division Multiple Access (TDMA)-based MAC protocol, in which the time domain is divided into minislots, and multiple minislots are grouped together to form a frame. Each frame is composed of a control subframe and a data subframe. The control subframe is used to exchange control messages. Data transmissions occur in the data subframe, which includes T minislots with fixed durations, and is further divided into an uplink[5] subframe and a downlink[6]

[5] the direction from SS (through RS) to BS

[6] the direction from BS (through RS) to SS

subframe with T^u and T^d minislots respectively. Unlike the PMP mode, T^u does not have to be the same as T^d in the mesh mode.

The WiMAX MAC protocol [1] supports both centralized and distributed scheduling. In this work, we focus on centralized scheduling, which is composed of two phases. In the first phase, each SS transmits an MSH-CSCH request (Mesh Centralized Scheduling) message carrying bandwidth request information to its parent node in the routing tree. Each non-leaf SS also needs to include bandwidth requests from its children in its own request message. In the second phase, the BS determines the bandwidth allocation for each SS based on all requests collected in the first phase and notify SSs by broadcasting an MSH-CSCH grant message along the routing tree. Subsequently, each SS computes the actual transmission schedule based on the bandwidth granted by the BS using a common scheduling algorithm. The time period for exercising these two stages is referred to as a scheduling period whose duration is a multiple of the frame duration, depending on the network size. In this research, we assume that inputs of problems, such as bandwidth requests, service priority do not change within a scheduling period.

Logic Link Control (LLC) Sublayer Model

The WiMAX protocol supports Quality of Service (QoS) at the LLC sublayer, where each transmission is connection oriented and several types of service levels are supported. The highest level is called Unsolicited Grant Service (UGS), in which a fixed bandwidth is allocated to a flow no matter whether it has data to send or not. The next level is real time Polling Service (rtPS), where the BS periodically polls the SS about how much data it needs to send, and tries to allocate bandwidth thereafter. The third level is extended real time Polling Service (ertPS), which is similar to rtPS but supports no data packets for some period, which is useful for silence suppression in VoIP. The fourth level is non-real time Polling Service (nrtPS), where a minimum bandwidth is allocated to the flow constantly, and the flow can request more bandwidth in case of burst data. The lowest level is Best Effort (BE), which does not guarantee any QoS, it is served only when the BS has free bandwidth after satisfying all higher level flows.

These QoS levels in the WiMAX protocol are complicated because they are not just different priorities for data flows. If only time scheduling is concerned, scheduling with such QoS levels can be solved in a step-by-step fashion, where UGS requests are scheduled first, and rtPS request are scheduled with remaining minislots, and so on until BE requests are scheduled. However, this step-by-step scheme cannot guarantee solutions with the same quality when time scheduling is jointly considered with channel assignment or antenna beam control, since different optimum solutions for a previous step will generate different problems for the next step and may not be able to result in an optimum solution for all steps. For this reason, we do not use these QoS levels defined in WiMAX protocol for scheduling. Instead, we simply consider data flows with different priority levels, and use the priority values to compute the objective function in the formalized problem.

Network Layer Models

We consider a static WRN with one Base Station (BS) and multiple Relay Stations (RSs) and Subscriber Stations (SSs). The network is considered "static" in the sense that all nodes have fixed positions in a sufficiently long time period.[7] These nodes will form a spanning tree rooted at the BS for routing. An SS can directly communicate

[7] At least much longer than the scheduling period.

with the BS if it is within the transmission range. Otherwise, RSs need to be used as relays to communicate with the BS in a multihop fashion.

In this work, we consider only uni-cast traffic between RS/SSs and the BS. All traffic between two SSs needs to go through the BS and will be split into two traffic flows. RSs have full protocol stacks, which means that they can store and forward data, instead of just repeating bits.[8] Also, RSs may generate traffic request as well, because besides SSs, there may exist nodes connected to RSs through networks other than the WRN itself.[9] An RS will aggregate the traffic from SSs and other RSs it relays in addition to the traffic generated by itself. Therefore, the problem at the network layer is to find a path for each RS/SS with traffic requests to the BS. All the paths will form a routing or spanning tree rooted at the BS, which is a subgraph of the communication graph.

[8] This is called "non-transparent" relay in the 802.16j working group draft recommendations.[2]

[9] For instance, an RS in the WRN may also act as the access point (AP) of an 802.11 WLAN.

INTERFERENCE DEGREE THEORY

Definition and Applications

The interference degree $\delta(e)$ of a link e is defined as the maximum number of links that interfere with e but do not interfere with each other. It characterizes the potential loss of system capacity if link e is scheduled, since at most $\delta(e)$ other links can be scheduled instead of the link e for the same resource. The interference degree δ of the entire network is the maximum interference degree among all links.

For example in Fig. 5, with interference range model, the two circles denote the interference ranges initiate from the transmitter and receiver of the red link, respectively. Except for those two green links, all links shown in the figure interfere with the red link. The 4 blue dash-dotted links do not interfere with each other, and consist of the largest non-interfering set. Therefore, the interference degree of the red link is 4 in this case.

It is worth to point out that the interference degree does *not* measure the severity of interference in a network. For instance, consider a network in which all links interfere with each other. It has extremely severe interference. Then consider another network in which each link interferes with one and only one other link. This network is on the other extreme with a very low interference level. Surprisingly, both networks have the same interference degree: 1.

The interference degree is not defined for networks without any interference.[1] However, such exceptional networks are both unrealistic and of little research interest.

[1] Any pairs of links do not interfere with each other in such a network, which means that no two links are incident, and thus the "network" is extremely disconnected.

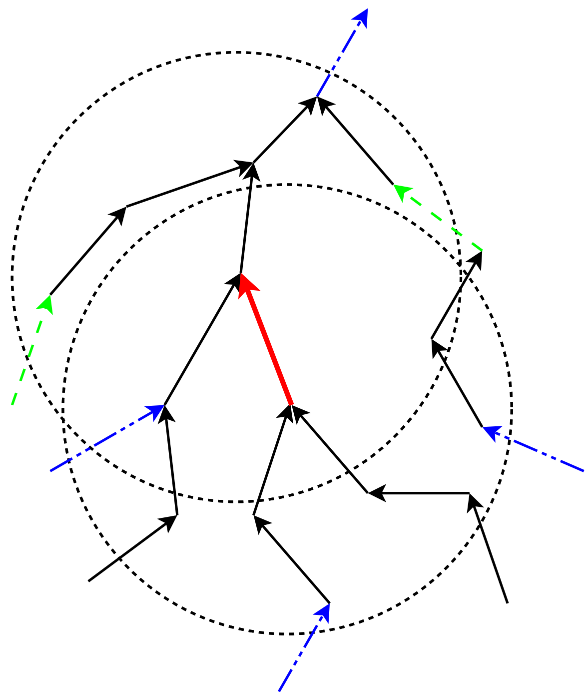


Figure 5: Illustration for the definition of interference degree

It has been shown by previous research that the performance of scheduling algorithms in wireless networks usually depends on the interference degree [36, 37, 38]. Researchers showed that maximal scheduling can guarantee to attain $\frac{1}{\delta}$ of the maximum throughput region in a network with an interference degree δ.

Preliminary Results

The interference degree of a link or a network is related to the interference model used. We have discussed various interference models in Chapter 3.

The most pessimistic model is the no sptial reuse model, where any pair of links is considered interfering with each other. Obviously, the interference degree of any link in such model is 1.

Other than this very special case, interference degree is rarely studied theoretically in the current literature.

In [36], the authors proved that the interference degree in a network with "bidirectional equal power model" is at most 8. The bidirectional equal power model they defined is essentially the same as the 1-hop interference model with the unit-disk transmission model we introduced in Chapter 3.

In the same paper, they showed that the interference degree can be unlimited in some networks with "unidirectional equal power model". The difference between their unidirectional equal power model and their bidirectional equal power model is that in the unidirectional model, all links are considered directional. This fits the WiMAX protocol better whereas the bidirectional model captures the nature of RTS-CTS based MAC protocols.

Additionally, they showed that the interference degree is at most 2, if the "node-exclusive spectrum sharing model" is applied, which is called primary interference model in this work.

One of the most important defects in [36] is that it did not consider the fact that the interference range is usually greater than the communication range, which directly affects the bound of interference degree in any network. Their bidirectional and unidirectional equal power models acutally assumed that the interference range equals the communication range.

On the other extreme, if the interference range is much greater than the communication range, then the transmitter node and the receiver node of a link can be considered a single point. Therefore, the interference degree problem becomes to decide how many points can be placed within a unit circle such that the distance between any two points is no more than 1. This is a well known problem with a tight bound of 7.[2]

In this work, we focus on the interference degree in WRN. We adopt the unidirectional link model and constrain the network topology to be a tree. Without loss of generality, we let the transmission range be 1 and the interference range be 2. All the theorems in this chapter still hold if the interference range is greater than 2.

Interference Degree of 2-hop WRN

First, we focus on the interference degree in any 2-hop WRN. We denote the BS as O,[3] the transmitters and receivers of each link on layer[4] 2 as T_i and R_i, respectively. Then we have $\|OT_i\| \leq 1$, $\|T_iR_i\| \leq 1$, and $1 < \|OR_i\| \leq 2$.

Lemma 1 *For any pair of non-interfering links (T_i, R_i) and (T_j, R_j) on layer 2, $\angle T_iOT_j > \frac{\pi}{3}$.*

Proof: Since $\|T_iR_j\| > 2$ and $\|T_jR_j\| \leq 1$,
in $\triangle T_iT_jR_j$, $\|T_iT_j\| > \|T_iR_j\| - \|T_jR_j\| > 1$.
In $\triangle OT_iT_j$, $\|OT_i\| \leq 1$, $\|OT_j\| \leq 1$, and $\|T_iT_j\| > 1$,
therefore $\angle T_iOT_j > \frac{\pi}{3}$. □

Lemma 2 *For any link (T_i, R_i) on layer 2, $\angle T_iOR_i < \frac{\pi}{3}$.*

Proof: In $\triangle OT_iR_i$, $\|OT_i\| \leq 1$, $\|OR_i\| > 1$, and $\|T_iR_i\| \leq 1$,
therefore $\angle T_iOR_i < \frac{\pi}{3}$. □

---

[2]If no two links should have exactly the same positions, then the bound would be 6.

[3]Without ambiguity, we use the same symbol to denote both the node and its position.

[4]We define layer-1 links as those links incident to the BS, layer-2 links as those links not incident to the BS but incident to a layer-1 link, etc. Some literature on 2-hop relay networks call the layer-1 links as *relay links* and the layer-2 links as *access links*.

Lemma 3 *For any pair of non-interfering links* (T_i, R_i) *and* (T_j, R_j) *on layer* 2, $\angle T_i O R_j > \frac{5\pi}{12}$.

Proof: In $\triangle O T_i R_j$, $\|OT_i\| \leq 1$, $\|OR_j\| \leq 2$, and $\|T_i R_j\| > 2$,

hence $\cos \angle T_i O R_j < \frac{\|OT_i\|^2 + \|OR_j\|^2 - \|T_i R_j\|^2}{2\|OT_i\| \cdot \|OR_j\|} < \frac{1}{4}$.

So $\angle T_i O R_j > \arccos \frac{1}{4} > \frac{5\pi}{12}$. □

Theorem 1 *The interference degree of any 2-hop WRN is at most 4.*

Proof: We prove this by showing that it is impossible to have 5 or more non-interfering links in any 2-hop WRN, which is a stronger proposition than the theorem.

First, we notice that for any two links at layer 1, they will interfere with each other. Second, for a link (O, T_1) at layer 1 and another link (T_2, R_2) at layer 2, they interfere with each other since $\|OR_2\| \leq 2$. Therefore, any two non-interfering links must be both at layer 2.

Assume that there exist 5 non-interfering links $(T_1, R_1), \ldots, (T_5, R_5)$ at layer 2. We denote $\theta(x)$ as the angle of the point x in the polar coordinate system originated at the BS O. Without loss of generality, let $\theta(T_1) \leq \cdots \leq \theta(T_5)$, as shown in Fig. 6.

We have $\theta(R_2) \geq \theta(T_1) + \angle T_1 O T_2 - \angle T_2 O R_2 > \theta(T_1) + \frac{\pi}{3} - \frac{\pi}{3} = \theta(T_1)$,

$\theta(T_3) = \theta(T_2) + \angle T_2 O T_3 > \theta(R_2) - \angle T_2 O R_2 + \angle T_2 O T_3 > \theta(R_2) - \frac{\pi}{3} + \frac{\pi}{3} = \theta(R_2)$, etc.

Therefore, $\theta(T_1) < \theta(R_2) < \theta(T_3) < \theta(R_4) < \theta(T_5)$.

According to Lemma 1 and 3, $\angle T_1 O R_2 + \angle R_2 O T_3 + \angle T_3 O R_4 + \angle R_4 O T_5 + \angle T_5 O T_1 > \frac{5\pi}{12} + \frac{5\pi}{12} + \frac{5\pi}{12} + \frac{5\pi}{12} + \frac{\pi}{3} = 2\pi$, which is impossible. This completes the proof. □

We show the interference degree bound 4 is tight by constructing an example in Fig. 7.

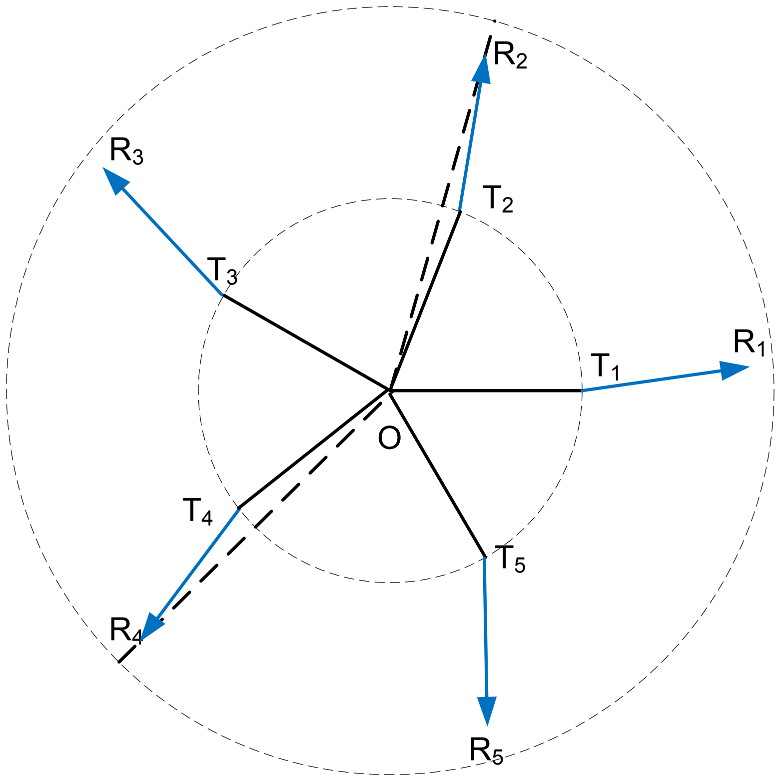


Figure 6: Impossible to have a 2-hop WRN with $\delta \geq 5$

The coordinates of the BS O is $(0,0)$, T_0 is $(0,-\epsilon')$, and R_0 is $(0,-1-\epsilon')$.[5] Therefore, (T_0,R_0) is a link at layer 2. The coordinates of T_1 to T_4 are $(0,-1+\epsilon)$, $(-1+\epsilon,0)$, $(0,1-\epsilon)$ and $(1-\epsilon,0)$ respectively. The coordinates of R_1 to R_4 are $(0,-2+\epsilon)$, $(-2+\epsilon,0)$, $(0,2-\epsilon)$ and $(2-\epsilon,0)$ respectively. Here ϵ and ϵ' are two arbitrarily small positive numbers and $\epsilon' < \epsilon$. It is easy to verify that links $(T_1,R_1),\ldots,(T_4,R_4)$ are all at layer 2. They do not interfere with each other but they all interfere with the link (T_0,R_0). Therefore, $\delta(T_0,R_0)=4$ and $\delta=4$ for this network.

[5]Out of the transmission range of the BS

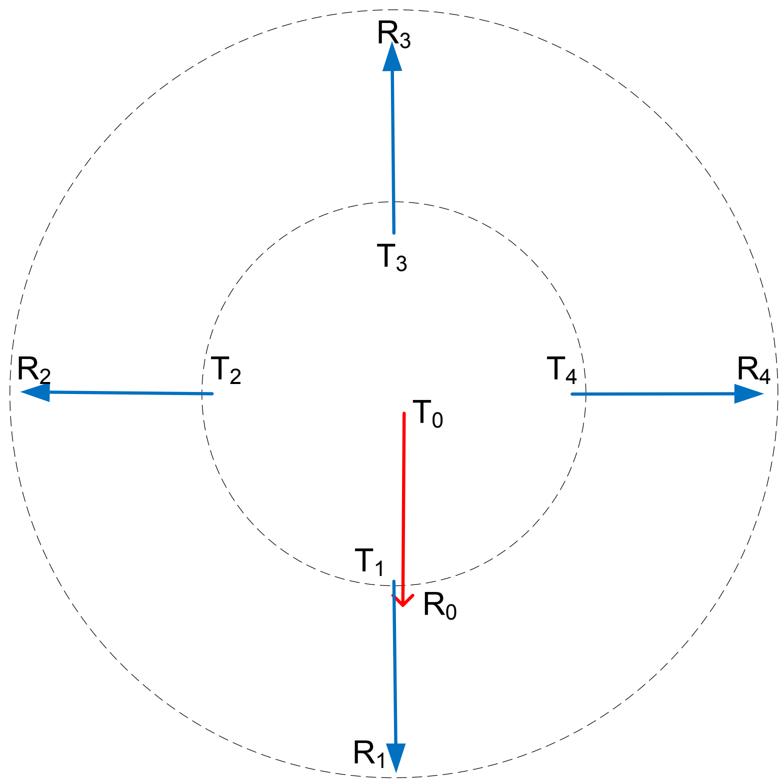


Figure 7: An example with $\delta = 4$

Interference Degree of General WiMAX Networks

Next we consider the interference degree in the general WiMAX networks.

Lemma 4 *For any pair of non-interfering links* (T_i, R_i) *and* (T_j, R_j)*, let their center points be* O_i *and* O_j *respectively. Then* $\|O_iO_j\| > \sqrt{3}$.

Proof: Set the x-y coordinate system as follows: (see Fig. 8)

Let the center point of O_i and O_j be the origin point and coordinates of O_i and O_j be $(0, d)$ and $(0, -d)$, respectively.

Let $l_i = \frac{1}{2}\|T_iR_i\|$, $l_j = \frac{1}{2}\|T_jR_j\|$, $\theta_i = \angle T_iO_iO_j - \frac{\pi}{2}$, and $\theta_j = \angle T_jO_jO_i - \frac{\pi}{2}$.

Then $T_i = (l_i \cos\theta_i, d + l_i \sin\theta_i)$,

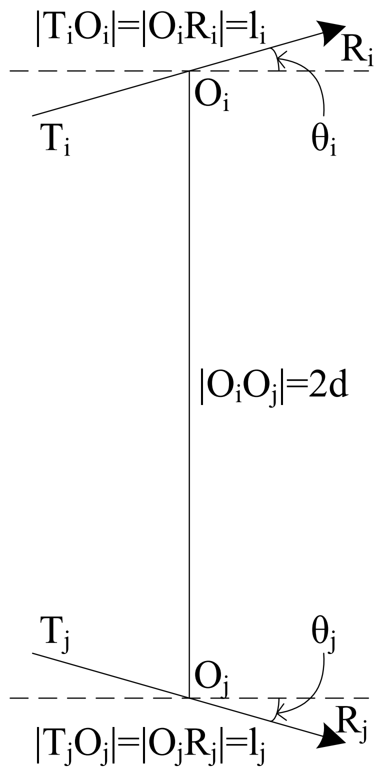


Figure 8: Demostration graph for the proof of Lemma 4

$R_i = (-l_i \cos\theta_i, d - l_i \sin\theta_i)$,

$T_j = (l_j \cos\theta_j, -d - l_j \sin\theta_j)$,

and $R_j = (-l_j \cos\theta_j, -d + l_j \sin\theta_j)$.

We have $\|T_iR_j\|^2 = l_i^2 + l_j^2 + 2l_il_j\cos(\theta_i+\theta_j) + 4d^2 + 4d(l_i\sin\theta_i - l_j\sin\theta_j) > 2^2 = 4$.

Since $l_i \leq \frac{1}{2}$, $l_j \leq \frac{1}{2}$, and $\cos(\theta_i+\theta_j) \leq 1$,

we can deduce that $4d^2 + 4d(l_i\sin\theta_i - l_j\sin\theta_j) > 3$.

Similarly, we can deduce that $4d^2 + 4d(l_j\sin\theta_j - l_i\sin\theta_i) > 3$ from $\|T_jR_i\| > 2$.

Therefore, $8d^2 > 6$, and $\|O_iO_j\| = 2d > \sqrt{3}$. □

Lemma 5 *For any pair of interfering links (T_i, R_i) and (T_j, R_j), let their center points be O_i and O_j respectively. Then $\|O_iO_j\| \leq 3$.*

Proof: Without loss of generality, we assume that the interference is caused by $\|T_iR_j\| \leq 2$. Then we have: $\|O_iO_j\| \leq \|O_iT_i\| + \|T_iR_j\| + \|R_jO_j\| \leq \frac{1}{2} + 2 + \frac{1}{2} = 3$. □

Theorem 2 *The interference degree of any WiMAX network is at most* 14.

Proof: Consider a link with center point O. Suppose there are m non-interfering links which interfere with this link. Denote the center points of these m links as $O_1, \ldots, O_m$. Then the center points of all the links satisfy:

$$\|O_iO_j\| > \sqrt{3}, \quad i, j \in \{1, \ldots, m\} \quad \text{by Lemma 4} \tag{1}$$

$$\|OO_i\| \leq 3, \quad i, j \in \{1, \ldots, m\} \quad \text{by Lemma 5} \tag{2}$$

If we draw a circle with diameter $\sqrt{3}$ centered at each O_i, these circles will not overlap. This is because any two overlapping circles with diameter $\sqrt{3}$ will have the distance between their center points less than $\sqrt{3}$, which contradicts with (1).

Next we consider a big circle with diameter $6 + \sqrt{3}$ centered at O. For any point P within any small circle centered at O_i, $\|O_iP\| \leq \frac{\sqrt{3}}{2}$ and $\|OO_i\| \leq 3$.[6] Therefore $\|OP\| \leq 3 + \frac{\sqrt{3}}{2}$, which means that all the small circles are contained in this big circle.

Kravitz has shown in [39] that $\frac{R}{r} > 4.52$ is a necessary condition for packing 15 circles of radius r in a larger circle of radius R.

But we have $\frac{6+\sqrt{3}}{\sqrt{3}} = 4.46\cdots < 4.52$.

Therefore, it is impossible to pack 15 such small circles in the big circle, which means it is also impossible to find 15 non-interfering links that all interfere with the same link. □

To decide the bound of the interference degree in an arbitrary WiMAX network is much more difficult. We proved a tight bound for the 2-hop special case, but we cannot prove the tightness for the bound 14 for the general case. However, it is

[6] from (2)

possible to construct an example with $\delta \geq 10$ as show in Fig. 9,[7] where the link (T, R) has 10 non-interfering links $(T_0, R_0), \ldots, (T_9, R_9)$ that all interfere with it. Therefore, the tight bound should be an integer in $[10, 14]$.

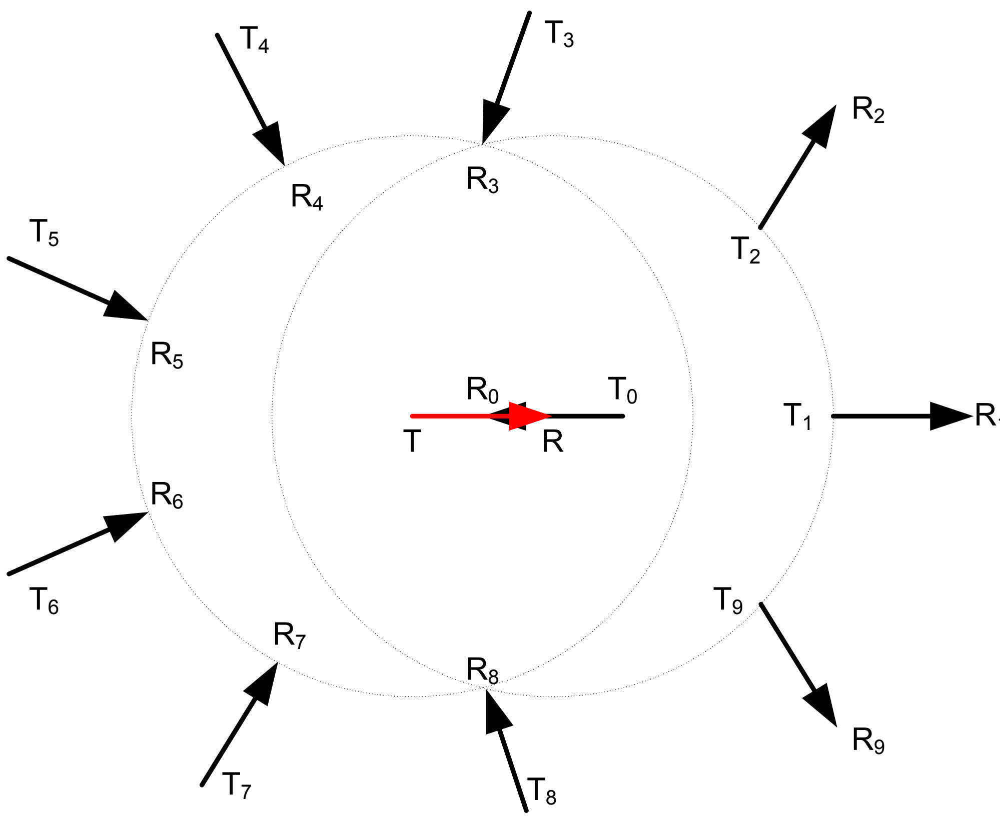


Figure 9: An example link with interference degree of at least 10

[7] All links have unit lengths. The distance from any node in $\{R_3, \ldots, R_8\}$ to T is 2. The distance from any node in $\{T_1, T_2, T_9, R_3, R_8\}$ to R is also 2.

INTERFERENCE AWARE ROUTING TREE CONSTRUCTION

We study the routing problem first since a routing tree is given as the input for a scheduling problem. It would be more precise to tell which tree is the best with bandwidth demand information. However, a routing tree is normally constructed beforehand and will be used for a relatively long period during which the traffic demands may change.

Problem Definition

It is well-known that interference has significant impacts on network performance [40]. So we will construct a low-interference tree, which can hopefully provide good throughput for any bandwidth demands. Note that both uplink and downlink traffic use the same tree for routing. Every link in the communication graph G is treated as an *undirectional* link for the tree construction.

If an SS/RS can directly connect to the BS or another RS closer to BS, we prefer not to use other RS(s) as relay(s) because additional hops will introduce longer delay, and more importantly, result in more severe interference. Therefore, once the communication graph G is given, we can easily determine on which layer of the routing tree should an SS/RS v_i appear, by conducting a Breadth First Search (BFS) on G.

We denote V_h and E_h as the set of nodes in layer h and the set of links between layer h and $(h-1)$ respectively. Moreover, $h_{\max}$ denotes the total number of layers (excluding v_0), i.e., the height of the tree. The tree construction problem is essentially to determine which node at layer $(h-1)$ should serve as the parent node for each node v_i at layer h. For all the RSs at the first layer, their parent must be the BS (v_0).

We differentiate the primary and secondary interference because the primary interference can only be resolved by scheduling, whereas the secondary interference may also be eliminated by carefully assigning DOFs. Given a tree, the primary interference value of a node v_i, $I^p(v_i)$, is defined as the total number of links incident to v_i on the tree. We define the secondary interference value of v_i as $I^s(v_i) = |N_i| - 1$, and the secondary interference value of link $e = (v_i, v_j)$ as $I^s(e) = \max\{I^s(v_i), I^s(v_j)\}$, where N_i is the set of nodes that can potentially interfere with v_i. In addition, we define a secondary interference bound for each layer $h > 1$, $I^b(h) = \max\{I^s(e_b), K-1\}$, where e_b is a bottleneck link, i.e., if all links with $I^s(e) \geq I^s(e_b)$ in G are removed, at least one node in V_h will be disconnected from nodes in V_{h-1}.

Definition 1 (ITCP) *The* **Interference aware Tree Construction Problem** *seeks a spanning tree Y rooted at the BS, such that in each layer $h > 1$, $I^p_{\max} = \max_{v \in V_h} I^p(v)$ is minimized subject to the constraint that $I^s(e) \leq I^b(h)$, $\forall e \in Y \cap E_h$.*

By solving ITCP, a balanced routing tree will be constructed. Moreover, potential secondary interference on this tree is likely possible to be eliminated by properly assigning DOFs.[1]

A Polynomial-Time Exact Algorithm

We present Algorithm 1, an optimal algorithm, to solve the ITCP. It's inputs include the communication graph G, an array H which gives the layer of each node, and $h_{\max}$, the number of layers.[2] An array *Parent* is the output, which specifies the parent node for each SS.

[1]Check Algorithm 6 in Chapter 6.

[2]G is treated as an undirectional graph for the tree construction. H and $h_{\max}$ can be computed using Breadth-First-Search.

Step 1 $V_1 \leftarrow \{v|H[v] = 1\}$;
forall $v \in V_1$
$Parent[v] \leftarrow v_0$;
endforall
forall $v \in V \setminus V_1$
$Parent[v] \leftarrow$ **null**;
endforall
$h \leftarrow h_{\max}$;

Step 2 **if** $h = 1$
return $Parent$;

Step 3 $V_h \leftarrow \{v|H[v] = h\}$;
$V_{h-1} \leftarrow \{v|H[v] = h - 1\}$;
$d_{\max} \leftarrow \max_{v \in V_{h-1}} d_v$, where d_v is the number of v's neighbors in V_h;
$lb \leftarrow 0$; $ub \leftarrow d_{\max}$;

Step 4 **while** $lb \leq ub$
$mid \leftarrow \lfloor \frac{lb+ub}{2} \rfloor$;
Construct the auxiliary graph $G' = (V', E')$;
Apply the Ford-Fulkerson algorithm on G' to find the maximum flow f from node s to node t and the corresponding link flow allocation $Flow$;
if $f = |V_h|$
$ub \leftarrow mid - 1$;
else
$lb \leftarrow mid + 1$;
endif
endwhile

Step 5 **forall** $e = (u, v) \in E'$
if $Flow[e] = 1$ **and** $u \neq s$ **and** $v \neq t$
$Parent[u] \leftarrow v$;
endif
endforall

Step 6 $h \leftarrow h - 1$;
goto Step 2;

Algorithm 1: Solve-ITCP$(G, H, h_{\max})$

Our algorithm constructs the tree in a bottom-up fashion. It starts from nodes at layer $h_{\max}$, and selects a node from nodes at layer $h-1$ as the parent node for each node at layer h in Steps 3–5. In Step 4, each node at layer h is connected to some node at layer $h-1$, while the maximum degree of nodes at layer $h-1$ is minimized using binary search. The auxiliary graph $G' = (V', E')$ in Step 4 is constructed as follows: $V' = V_h \cup V_{h-1} \cup \{s, t\}$, where s and t are the auxiliary source and sink nodes respectively. For each $v \in V_h$, we create a directional link with a capacity of 1 from s to v. For each $u \in V_h$ and $v \in V_{h-1}$, we create a directional link e with capacity 1 from u to v, if $(u, v) \in E$ and $I^s(e) \leq I^b(h)$. Finally, for each $v \in V_{h-1}$, we create a directional link with a capacity of mid from v to t. In Step 5, we compute the parent assignment for nodes at layer h according to the link flow allocation $Flow$. We use a simple example to illustrate the construction of the auxiliary graph in Fig. 10. In

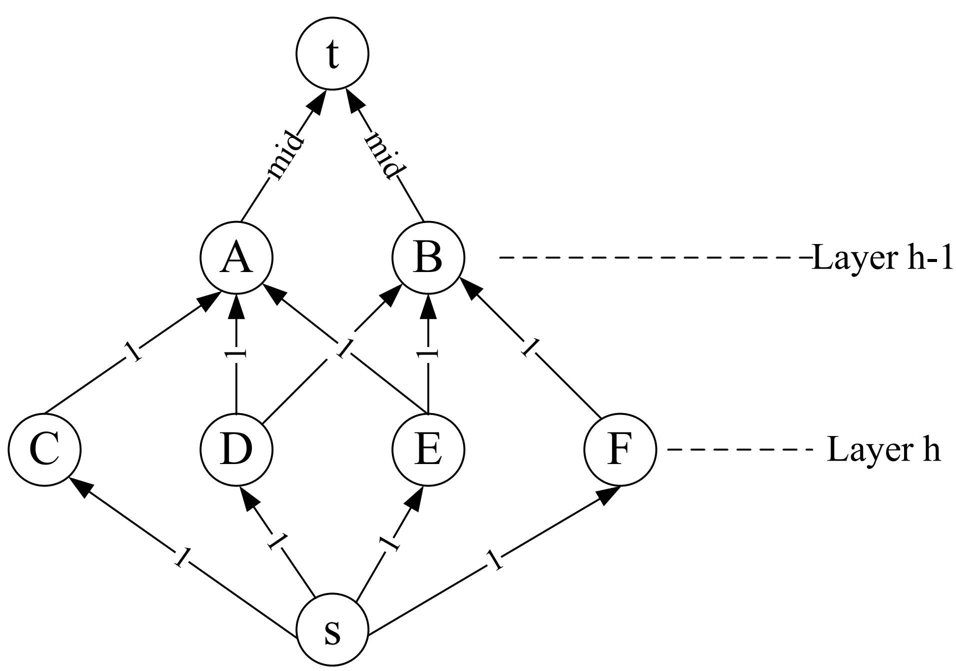


Figure 10: The auxiliary graph G'

the figure, the secondary interference values of links (C, A), (D, A), (E, A), (D, B), (E, B) and (F, B) are assumed to be no more than the corresponding bound $I^b(h)$.

Theorem 3 *Algorithm 1 optimally solves ITCP in $O(mnh_{\max} \log \delta_{\max})$ time, where m, n, $h_{\max}$ and $\delta_{\max}$ are the number of links, the number of nodes, the number of layers and the maximum node degree of G respectively.*

Proof: As mentioned before, ITCP is essentially the problem of determining which node should be selected as the parent node for each node v at the next layer. Since the secondary interference value of each link crossing two layers can be predetermined, the constraint of the ITCP can be satisfied by including only links with $I^s(e) \leq I^b(h)$ in the auxiliary graph. Therefore, at each layer h, the problem boils down to determine a parent node assignment in the bipartite graph given by V_{h-1}, V_h and the links in between, such that each node in V_h is connected to exactly one node in V_{h-1} and the maximum degree of nodes in V_{h-1} is minimized. Note that the primary interference value of a non-leaf node actually equals its degree. In each iteration of the binary search in Step 4, we need to check if there exits an assignment such that each node in V_h is connected to exactly one node in V_{h-1} and the degrees of each node in V_{h-1} is no more than *mid*. Next, we show that there exists such an assignment if and only if the maximum s–t flow in G' equals $|V_h|$.

First, it is well-known that the augmenting path based maximum flow algorithm such as the Ford-Fulkerson algorithm can always find a maximum flow whose corresponding link flows are all integers if the capacity of each link is an integer. If the maximum flow found by the Ford-Fulkerson algorithm is $|V_h|$, then there must be exactly one unit flow going into each node in V_h from s since the capacity of every link between s and a node $v \in V_h$ is 1. According to the flow conservation constraint and integer flow claim mentioned above, there must be exactly one unit flow going from every node V_h to a node in V_{h-1}, which actually leads to a feasible parent node assignment. Moreover, the capacities of the links connecting nodes in V_{h-1} to t are set to mid, which ensures that based on the assignment, each node in V_{h-1} has no more than mid children.

In Algorithm 1, Step 1 takes $O(n)$ time for initialization. The time complexity of Step 3 depends on the number of nodes and links at the two consecutive layers, which are obviously bounded by m and n respectively. Therefore, Step 3 takes $O(m + n)$ time. The Ford-Fulkerson algorithm can find the maximum flow in $O(|E'|f_{\max})$ time, where $f_{\max} \leq |V_h|$ for our problem. Moreover, $d_{\max}$ is bounded by $\delta_{\max} - 1$, where $\delta_{\max}$ is the maximum node degree in the communication graph G. So Step 4 can be done in $O(\log(\delta_{\max} - 1)|V_h||E'|) = O(mn \log \delta_{\max})$ time. It is easy to see that Step 5 takes $O(|E'|) = O(m)$ time. Steps 3–5 are executed $h_{\max} - 1$ times. The total time complexity of Algorithm 1 is therefore $O(mnh_{\max} \log \delta_{\max})$. □

Our tree construction algorithm runs very fast in practice because the number of links between any two consecutive layers is usually much less than m, and both $h_{\max}$ and $\delta_{\max}$ are normally small.

JOINT SCHEDULING AND DOF ASSIGNMENT

After we solved the routing tree construction problem as described in the last chapter, we are ready to study the link-level resource allocation problems. In this work, we consider three types of resources to allocate: the time, the spectrum, and the signal power. Specifically, time resource means all minislots in the frame; spectrum resource is the orthogonal channels; and the signal power resource is represented by DOF assignment (controling the antenna pattern) at each node. It is not easy to optimally manage even one of these 3 resources. Furthermore, the allocation of these 3 resources are interdependent with each other. Therefore, it is extremely hard to solve the resource allocation problem considering all 3 resources together.

We will first ignore the channel assignment and emphasize joint scheduling and DOF assignment. We consider a WRN in which each node has a single half-duplex radio and a DAA antenna with K DOFs. All transmissions are conducted on a single common channel. We model the network using a *communication graph* $G = (V, E)$, where $V = \{v_0, v_1, \ldots, v_{n-1}\}$. v_0 corresponds to the BS, and $v_1, \ldots, v_{n-1}$ are RS/SSs. The edge set E can be decided using the arbitrary graph model we introduced in Chapter 3, which is the most generalized model among all transmission models. We adopt one of the most generalized model—the protocol interference model in Chapter 3, in which the potential interference sources to a link is a predetermined set.

Problem Definition

The scheduling problem considered here is different from that in a network with omni-directional antennas because DOFs can be used to suppress interference. Therefore, our scheduling problems involve both transmission scheduling and DOF assignment. The input of the scheduling problem includes a spanning tree with the BS v_0 as the root, $n-1$ RS/SSs $\{v_1, \ldots, v_{n-1}\}$, their bandwidth demands $\mathbf{Q^u} = [q_1^u, \ldots, q_{n-1}^u]$ and $\mathbf{Q^d} = [q_1^d, \ldots, q_{n-1}^d]$ for uplink and downlink respectively, and the uplink/downlink subframe sizes T^u/T^d. q_i^u indicates the number of minislots v_i needs to transmit its uplink traffic. Note that if v_i is an RS, q_i^u includes the bandwidth needed for itself but does not include the bandwidth requested by any of its descendants on the tree.

We define an uplink scheduling matrix $\mathbf{\Gamma}$ and a DOF assignment matrix $\mathbf{\Lambda}$. $\mathbf{\Gamma}_i^t = 1$ if link (v_i, v_{p_i}) is active in minislot t; $\mathbf{\Gamma}_i^t = 0$ otherwise.[1] $\mathbf{\Lambda}_{i,j}^t = 1$ if v_i assigns a DOF to point at v_j for communications or interference supression in minislot t; $\mathbf{\Lambda}_{i,j}^t = 0$ otherwise.

$\mathbf{\Lambda}$ has nothing to do with DOFs assigned for communications since $\mathbf{\Gamma}_{i,j}^t = 1$ implies that one DOF at v_i needs to be assigned for transmission and one DOF at v_j needs to be assigned for reception.

A scheduling matrix $\mathbf{\Gamma}$ and a DOF assignment matrix $\mathbf{\Lambda}$ are said to be *feasible* if:

1. $\forall i, t, \mathbf{\Lambda}_{i,p_i}^t \cdot \mathbf{\Lambda}_{p_i,i}^t \geq \mathbf{\Gamma}_i^t$ (for each active link, DOFs need to be assigned at both ends for communication);

2. there does not exist primary or secondary interference in any minislot;

3. $\forall i, t, \sum_{j=0}^{n-1} \mathbf{\Lambda}_{i,j}^t \leq K$ (each node has only K DOFs).

We also define an uplink bandwidth allocation vector $\mathbf{B^u} = [b_1^u, \ldots, b_{n-1}^u]$, its corresponding satisfaction ratio vector $\mathbf{S^u} = [s_1^u, \ldots, s_{n-1}^u] = [\frac{b_1^u}{q_1^u}, \ldots, \frac{b_{n-1}^u}{q_{n-1}^u}]$,[2] and its corresponding aggregated bandwidth allocation vector $\mathbf{A^u} = [a_1^u, \ldots, a_{n-1}^u]$, where b_i^u

[1]Only $n-1$ links on the given tree will be considered for scheduling.

[2]We define the satisfaction ratio of $\frac{0}{0} = 1$.

indicates the actual bandwidth (the number of minimslots in each frame) allocated to v_i for uplink traffic generated at v_i, and a_i^u indicates the actual bandwidth allocated to v_i for uplink traffic generated at v_i and all of its descendants. A bandwidth allocation vector $\mathbf{B^u}$ is said to be *feasible* if there exists a feasible scheduling matrix $\mathbf{\Gamma}$, such that $\forall i \in \{1, \ldots, n-1\}, \sum_{t=1}^{T^u} \mathbf{\Gamma}_i^t \geq a_i^u$.

Based on a scheduling matrix $\mathbf{\Gamma}$, the sustainable data rate on link $e_i = (v_i, v_{p_i})$ is $b(e_i) = \sum_{t=1}^{T^u} \mathbf{\Gamma}_i^t$.

Definition 2 (USP) *The* **Uplink Scheduling Problem** *seeks a feasible uplink bandwidth allocation vector* $\mathbf{B^u}$ *and its corresponding satisfaction ratio vector* $\mathbf{S^u}$*, along with a corresponding feasible scheduling matrix* $\mathbf{\Gamma}$ *and DOF assignment matrix* $\mathbf{\Lambda}$*, such that the minimum satisfaction ratio* $\min_{1 \leq i \leq n-1} s_i^u$ *is maximized.*

In the USP, we try to maximize the minimum satisfaction ratio for the fairness purpose. Similarly, we can define the Downlink Scheduling Problem (DSP). Note that if there exists an algorithm which can optimally solve the USP/DSP, then it can tell if a bandwidth demand vector can be fully satisfied ($\min_{1 \leq i \leq n-1} s_i^u = 1$) or not ($\min_{1 \leq i \leq n-1} s_i^u < 1$).

In this chapter, we will present algorithms to solve the scheduling problems. Since the uplink and downlink traffic are scheduled in different subframes according to the WiMAX MAC protocol [1], we will only discuss the USP and the corresponding algorithms in the following. The downlink scheduling simply follows.

The link scheduling problems in a multihop wireless network (even with omni-directional antennas) are usually NP-hard [12, 13]. Therefore, in the first part of this chapter, we consider a *special case* of the USP, where each node has a relatively large number of DOFs but a relatively small number of potential interferers in its neighborhood, such that there exists a DOF assignment which can eliminate all potential secondary interference.

For example, if the number of DOFs in each node v_i, $K \geq \lceil \frac{N_{\max}}{2} \rceil + 1$, where $N_{\max} = \max_{0 \leq i \leq n-1} |N_i|$, then there exists a trivial secondary interference free DOF assignment since half of total secondary interference can be taken care of by DOFs in the active receivers and another half can be dealt with by DOFs in the active transmitters. Therefore, in this special case, only the impact of the primary interference needs to be addressed for transmission scheduling.

In the second part, we propose a heuristic algorithm for the general case where both the primary and the secondary interference are addressed.

A Polynomial-Time Exact Algorithm for a Special Case

In the special case, a bandwidth allocation vector $\mathbf{B}$ is feasible if for its corresponding aggregated bandwidth allocation vector $\mathbf{A}$,

$$\forall v, \sum_{i \in D_v} a_i \leq T \text{ ,where } D_v = \{i | v_i = v \text{ or is a descendant of } v\}. \tag{3}$$

Therefore, it suffices to find a bandwidth allocation vector $\mathbf{B}$ with constraints in (3).

The basic idea of the proposed algorithm is to identify the bottleneck node in each step, compute the corresponding bandwidth allocation for both the bottleneck node and its descendants based on their demands, and then remove them from the tree. This procedure will be repeated until all the nodes are removed from the tree. The algorithm for solving the special case USP is formally presented as Algorithm 2, whose inputs include the bandwidth demand vector $\mathbf{Q^u} = [q_1^u, ..., q_{n-1}^u]$, the number of minislots available for uplink traffic T^u and the routing tree Y.

Step 1 $P \leftarrow \{i|v_i \text{ is an RS on } Y\}$;
$Z \leftarrow \{0, 1, \cdots, n-1\}$;
forall $i \in P$
$T_i \leftarrow T^u$;
endforall

Step 2 $\gamma_0 \leftarrow$ Schedule-BS($\mathbf{Q^u}, T_0$);
forall $i \in P \setminus \{0\}$
$\gamma_i \leftarrow$ Schedule-RS($\mathbf{Q^u}, T_i, i$);
endforall
$j \leftarrow \text{argmin}_{i \in P} \gamma_i$;

Step 3 $D \leftarrow \{i|v_i \text{ is a descendant of } v_j \text{ on } Y\}$;
$C \leftarrow \{i|v_i \text{ is an ancestor of } v_j \text{ on } Y\}$;
$B_j \leftarrow \sum_{i \in D} b_i$;
forall $i \in C$
$T_i \leftarrow T_i - B_j$;
endforall
$Z \leftarrow Z \setminus \{j\} \setminus D$;
if $Z \neq \emptyset$
goto Step 2;

Algorithm 2: Solve-Special-USP($\mathbf{Q^u}, T^u, Y$)

In Step 1 of Algorithm 2, we initiate the number of free minislots at all non-leaf nodes to T^u.[3] The algorithm starts with the BS and check the RS/SSs one by one to find the bottleneck node using Algorithms 3 and 4 in Step 2. The details will be discussed later. In Step 3, the bottleneck node and all of its descendants are removed from the tree, and the numbers of free minislots in all of its ancestors are updated. The procedure is repeated until all nodes are scheduled.

Since for all nodes other than the bottleneck node v_b, we have found scheduling schemes in Step 2 such that $\forall k, \gamma_k \geq \gamma_b$. We guarantee that other bandwidth demands scheduled later will have satisfaction ratios greater than or equal to γ_b.

[3]There is no need to consider any SS, since the scheduling scheme for its parent node contains the minislot allocation for the link in between, which is the only link incident to the SS.

Step 1 $\mathbf{B} \leftarrow \mathbf{0}$;
$q_{\text{total}} \leftarrow \sum_{1 \le i \le n-1} q_i$;
if $q_{\text{total}} \le T_0$
  $\mathbf{B} \leftarrow \mathbf{Q}$;
  return 1;
endif

Step 2 $\gamma \leftarrow \frac{T_0}{q_{\text{total}}}$;
forall $k \in \{1, \ldots, n-1\}$
  $b_k \leftarrow \lfloor \gamma q_k \rfloor$;
$s_k \leftarrow \frac{b_k}{q_k}$; ($s_k \leftarrow 1$ if $q_k = 0$)
endforall
$T_{\text{rem}} \leftarrow T_0 - \sum_{1 \le k \le n-1} b_k$;

Step 3 $j \leftarrow \text{argmin}_{1 \le k \le n-1} s_k$;
if $T_{\text{rem}} = 0$
  return s_j;
endif
$b_j \leftarrow b_j + 1$;
$s_j \leftarrow \frac{b_j}{q_j}$;
$T_{\text{rem}} \leftarrow T_{\text{rem}} - 1$;
goto Step 3;

Algorithm 3: Schedule-BS($\mathbf{Q}, T_0$)

Step 1 $\mathbf{B} \leftarrow \mathbf{0}$;
$D \leftarrow \{j | v_j \text{ is a descendant of } v_i \text{ on } Y\}$;
$D' \leftarrow D \cup \{i\}$;
$q_{\text{total}} \leftarrow q_i + 2\sum_{k \in D} q_k$;
if $q_{\text{total}} \leq T_i$
  forall $k \in D'$
    $b_k \leftarrow q_k$;
  endforall
  return 1;
endif

Step 2 $\gamma \leftarrow \frac{T_i}{q_{\text{total}}}$;
forall $k \in D'$
  $b_k \leftarrow \lfloor \gamma q_k \rfloor$; $s_k \leftarrow \frac{b_k}{q_k}$; ($s_k \leftarrow 1$ if $q_k = 0$)
endforall
$T_{\text{rem}} \leftarrow T_i - b_i - 2\sum_{k \in D} b_k$;

Step 3 $j \leftarrow \text{argmin}_{k \in D'} s_k$;
if $T_{\text{rem}} = 0$ **or** ($T_{\text{rem}} = 1$ **and** $b_i = q_i$)
  return s_j;
endif
if $j = i$ **or** $T_{\text{rem}} = 1$
  $b_i \leftarrow b_i + 1$;
$s_i \leftarrow \frac{b_i}{q_i}$;
$T_{\text{rem}} \leftarrow T_{\text{rem}} - 1$;
else
  $b_j \leftarrow b_j + 1$;
$s_j \leftarrow \frac{b_j}{q_j}$;
$T_{\text{rem}} \leftarrow T_{\text{rem}} - 2$;
endif
goto Step 3;

Algorithm 4: Schedule-RS($\mathbf{Q}, T_i, i$)

Algorithms 3 and 4 are similar. They not only test whether the BS or a particular RS is the bottleneck node, but also compute a corresponding bandwidth allocation.[4] In both algorithms, q_{total} gives the total number of minislots required for the traffic that needs to go through the corresponding node.[5] At every RS v_i (including the BS v_0), if $q_{\text{total}} \leq T_i$, then both its demand and its descendants' demands can be fully satisfied. Otherwise, the bandwidth is allocated to nodes in the subtree rooted at v_i according to the ratio $\frac{q_{\text{total}}}{T^u}$. After that, the remaining minislots (if there are any) will be allocated to nodes in the ascending order of their current satisfaction ratios, until no more allocation can be made. After such an allocation procedure, the minimum satisfaction ratio of nodes on the subtree rooted at v_i can be obtained and returned, which we call *the effective satisfaction ratio of* v_i. The RS with the minimum effective satisfaction ratio will be identified as the bottleneck node.[6] Basically, the nodes in the upper layers are more likely to be the bottleneck node since they need to handle more relay traffic.

We present Algorithm 3 for the BS and Algorithm 4 for the RSs since the bandwidth allocation in the BS is different from that in an RS. An RS v_i needs to allocate bandwidth for traffic generated by itself as well as relay traffic generated by its descendants. Therefore, in order to provide one minislot to one of its descendants v_k, two minislots need to be arranged for v_k at v_i, one for link (v_{h_i}, v_i) and another for link (v_i, v_{p_i}), where h_i and p_i are the indices of the child to relay v_k's traffic and the parent node of v_i respectively. However, the bandwidth allocation in the BS is simpler since it does not have any parent node.

---

[4]Once the bandwidth allocation is determined, it is trivial to find a corresponding transmission schedule and DOF assignment in the special case.

[5]Including the traffic generated by itself and all its descendants.

[6]The number of minislots that can allocate for a node v_k at its different ancestors may be different.

Algorithm 3 seeks a scheduling scheme at the root node BS, updates the bandwidth allocation vector $\mathbf{B}$, and returns the minimum satisfaction ratio of all demands. In Step 1, we sum up all the demands and satisfy them if the available minislots are enough. Otherwise, we scale down all demands by γ and count the remaining free minislots N in Step 2. Then we keep allocating one minislot to the demand with lowest satisfaction ratio in Step 3 until all free minislots are depleted.

Similar to Algorithm 3, Algorithm 4 seeks a bandwidth allocation at an RS v_i with available minislots T_i and bandwidth demands $\mathbf{Q}$, updates part of the bandwidth allocation vector $\mathbf{B}$, and returns the effective satisfaction ratio of all demands initiated from the subtree rooted at this intermediate node.

We will use an example to demonstrate how Algorithms 2, 3 and 4 solve the scheduling problem in the special case. The example network is shown in Fig. 11.

In Fig. 11, we denote the BS as v_0, the two RSs as v_1 and v_2, and the leaf level SSs as $v_3, \ldots, v_7$ (all from left to right). Let bandwidth requests of each RS/SS be $\mathbf{Q} = [1, 1, 2, 3, 2, 3, 4]$, and the number of minislots in a scheduling period be $T = 16$.

First, we need to find out the bottleneck node in the network, which has the lowest effective satisfaction ratio. We accumulate all bandwidth requests through v_0 and get $Q_{\text{total}} = 16 \leq T$. This shows that all requests can be satisfied from v_0's point of view. And therefore its effective satisfaction ratio is 100%. Similarly, we pre-schedule for v_1 and find out that v_1 is not the bottleneck node either. However, when we next pre-schedule for v_2, we get $Q_{\text{total}} = 19 > T$. This shows that not all requests through v_2 can be satisfied. We first allocate $\frac{16}{19}$ of the requested minislots to v_2, v_5, v_6 and v_7, rounding down if necessary. Accordingly, the four nodes get 0, 1, 2 and 3 minislots with satisfaction ratios 0, $\frac{1}{2}$, $\frac{2}{3}$ and $\frac{3}{4}$ respectively, and we have 4 free minislots left. Then we allocate those 4 minislots, starting from the most unsatisfied node v_2, then v_5. Note that in order to allocate one more minislot to v_5, two minislots need to be

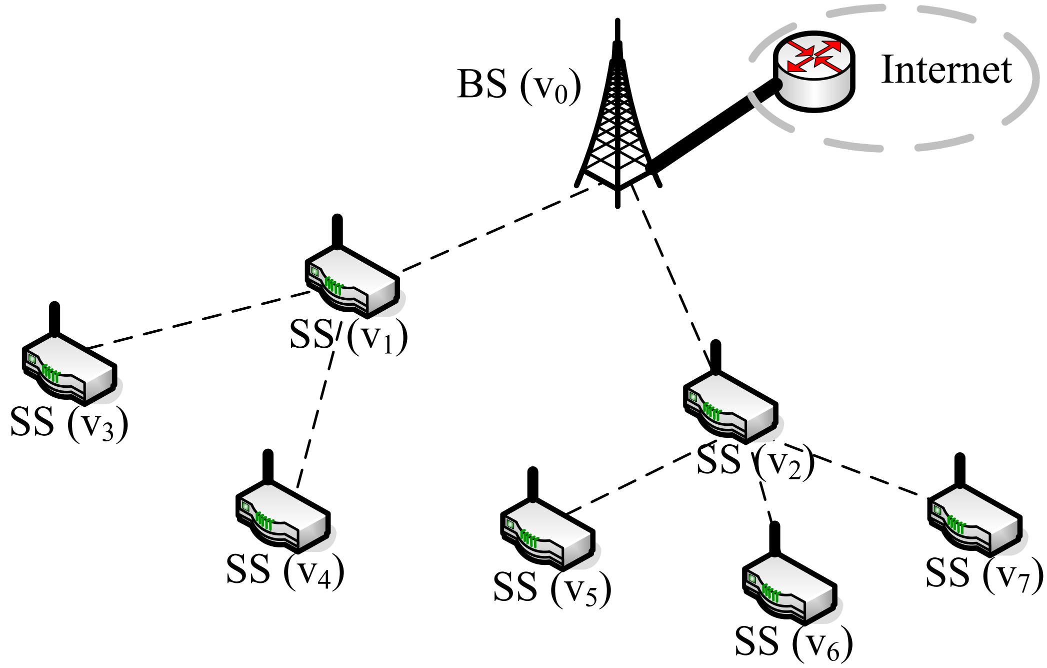


Figure 11: An example WiMAX network

consumed at v_2 because v_2 needs to receive data from v_5 and then forward them to v_0. Therefore, after allocating minislots to v_5, there is only one minislot left and no more allocation for v_6 or v_7 is possible. We end up with the four nodes each allocated 1, 2, 2 and 3 minislots, with satisfaction ratios 1, 1, $\frac{2}{3}$ and $\frac{3}{4}$. The effective satisfaction ratio for v_2 is $\frac{2}{3}$, the lowest of the four ratios. Therefore, v_2 is the bottleneck node.

After scheduling for all requests through v_2, we remove the subtree rooted at v_2 and all its requests from the graph, locate the next bottleneck node and schedule for the remaining tree recursively. In this example, the remaining requests can all be fully satisfied. The final allocation vector for the entire graph is $\mathbf{B} = [1, 1, 2, 3, 2, 2, 3]$. Except for nodes v_6 and v_7, all nodes' bandwidth requests are satisfied.

Theorem 4 *Algorithm 2 computes a bandwidth allocation vector* $\mathbf{B}$ *with max-min satisfaction ratio in* $O(n^3 \log n)$ *time.*[7]

Proof: First of all, we show that Algorithm 3 always computes a bandwidth allocation vector $\mathbf{B}$ with max-min satisfaction ratio based on the given minislot availability. If all bandwidth demands can be 100% satisfied, Algorithm 3 terminates at Step 1 and can obviously find a bandwidth allocation vector $\mathbf{B}$ with max-min satisfaction ratio. Otherwise, the algorithm terminates when the number of remaining free minislots $T_{rem} = 0$. In this case, if there exists another bandwidth allocation vector $\mathbf{B}'$ with a larger minimum satisfaction ratio, i.e., $s'_{\min} > s_{\min}$, then $\exists k, b'_k \geq b_k + 1$. Since there is no free minislots left, increasing the bandwidth allocation of some node v_k must lead to decreasing the bandwidth allocation of another node v_j, i.e., $\exists j, b'_j \leq b_j - 1$. Therefore, we have

$$s'_j = \frac{b'_j}{q_j} \leq \frac{b_j - 1}{q_j} = \frac{\lfloor \gamma q_j \rfloor - 1}{q_j} \leq \gamma - \frac{1}{q_j} \tag{4}$$

In Step 2 of Algorithm 3, since each b_k is rounded down to $\lfloor \gamma q_k \rfloor$, $\forall k, (\gamma - s_{\min}) q_k \leq 1$. Therefore, $\gamma - \frac{1}{q_j} \leq s_{\min}$. Combining this with (4), we have $s'_j \leq s_{\min}$. This contradicts with the assumption that $s'_{\min} > s_{\min}$. Therefore, there does not exist a bandwidth allocation vector $\mathbf{B}'$ with a larger minimum satisfaction ratio, i.e., $s'_{\min} > s_{\min}$. Similarly, we can prove that Algorithm 4 also computes a bandwidth allocation vector $\mathbf{B}$ with max-min satisfaction ratio.

In Step 2 of Algorithm 2, if BS is the bottleneck node, i.e., $j = 0$, we will simply get the bandwidth allocation vector from Algorithm 3, which has been shown above to

[7]The input for scheduling problems are trees. Therefore $O(m) = O(n)$.

have the max-min satisfaction ratio. Next, we consider the case when the bottleneck node v_j is an RS, i.e., $j \neq 0$. Let $s_{\min} = \min_{1 \leq i \leq n-1} s_i$ be the minimum satisfaction ratio found by Algorithm 2. Since v_j is the bottleneck node, $s_{\min} = s_j$. It is impossible to find another bandwidth allocation vector $\mathbf{B}'$ with minimum satisfaction ratio $s'_{\min} > s_{\min}$. Otherwise the effective satisfaction ratio of v_j in $\mathbf{B}'$: $s'_j \geq s'_{\min} > s_{\min} = s_j$. And this contradicts with our proof that s_j is maximized for the subtree rooted at v_j.

Next, we show that Algorithm 2 is a polynomial-time algorithm. In Algorithm 3, both Step 1 and Step 2 take $O(n)$ time. Step 3 takes $O(n \log n)$ time to process s_k in order. Therefore, Algorithm 3 takes $O(n \log n)$ time. Similarly, Algorithm 4 can be done in $O(n \log n)$ time. In Algorithm 2, it takes $O(n)$ time for initialization in Step 1. In Step 2, Algorithm 4 is executed $O(n)$ times, each of which takes $O(n \log n)$ time. Hence, the total running time of this step is $O(n^2 \log n)$. It takes $O(n)$ time for updating in Step 3. Step 3 removes at least one node from the spanning tree Y. The loop composed of Step 2 and Step 3 will be executed for $O(n)$ times. Therefore, the time complexity of Algorithm 2 is $O(n^3 \log n)$. □

In most cases, the bottleneck node is either the BS or an RS in the first layer. Therefore, Step 2 of Algorithm 2 will only be executed a few times. And the running time of Algorithm 2 is only $O(n^2 \log n)$ in practice.

An Algorithm for General Cases

We present an efficient heuristic algorithm (Algorithm 5) to solve the USP in the general case. It includes a subroutine that can optimally determine whether a set of links can be active simultaneously, which is not trivial in the context of DAA antennas since DOFs can be allocated to suppress interference and enable concurrent transmissions.

```
Step 1 Γ ← 0;
       Λ ← 0;
       B ← 0;
       t ← 1;
       forall i ∈ {1, ··· , n − 1}
         a_i ← Σ_{k∈D_i} q_k + b_i;
         x_i ← 0;
       endforall

Step 2 Sort L^s in the ascending order of link satisfaction ratios;
       L ← ∅;

Step 3 forall e = (v_i, v_j) ∈ L^s
         A ← AssignDOF(L, e);
         if A ≠ ∅
           L ← L ∪ {e};
           Γ_i^t ← 1;
           b_i ← b_i + 1;
           x_i ← x_i + 1;
           if x_i = a_i
             L^s ← L^s \ {e};
           endif
           forall (k, l) ∈ A
             Λ_{k,l}^t ← 1;
           endforall
         endif
       endforall

Step 4 t ← t + 1;
       if t ≤ T^u
         goto Step 2;
```

Algorithm 5: Solve-USP($\mathbf{Q}, T^u, L^s$)

Algorithm 5 computes a scheduling matrix $\mathbf{\Gamma}$ with a scheduling period of T^u minislots, a DOF assignment $\mathbf{\Lambda}$ and a corresponding bandwidth allocation vector $\mathbf{B}$ for the given bandwidth demands $\mathbf{Q}$ and a set L^s of links on the routing tree Y. This is a greedy algorithm which tries to pack as many links as possible in a minislot in the ascending order of the satisfaction ratios of all links. Note that the satisfaction ratio of a link is equal to the number of minislots which have been allocated to it divided by the total number of minislots needed for transmitting both local and relay traffic. The core part of this algorithm is the subroutine AssignDOF (Algorithm 6), which determines whether a set of links can be active concurrently and gives a feasible DOF assignment if the answer is *yes*.

In essence, Algorithm 5 greedily tries to maximize the minimum satisfaction ratio of all aggregated bandwidth demands. It activates individual links in each minislot instead of allocating minislots to individual bandwidth demands, which may need multi-hop transmissions. Therefore, the algorithm may activate a link before activating links of its ancestors. If a link has less minislots allocated than all the minislots allocated to its children links, then some bandwidth will be wasted. However, this mis-allocation is largely avoided by always trying to activate links at lower layers first.

Algorithm 6 checks whether a link e can be active simultaneously with the link set L. In Step 1, we check whether there exists primary interference between e and some link in L. If so, there is no way to activate them concurrently via DOF assignment. Moreover, if there is no secondary interference between e and any link in L,[8] we immediately know e can be active concurrently with L. Otherwise, an auxiliary directional graph $G' = (V', E')$ is constructed to find a feasible DOF assignment, which consists of three types of vertices. In the following, we use V_T and V_R to denote the set of

[8] Includes the case when $L = \emptyset$

Step 1 **if** $\exists l \in L, e$ is incident to l
  return $\emptyset$;
endif

if $e = (v_i, v_j)$ does not have secondary interference with any link in L
  return $\{(i,j),(j,i)\}$;
endif

Step 2 Construct the auxiliary graph $G' = (V', E')$;
Apply the Ford-Fulkerson algorithm on G' to find a maximum flow $f_{\max}$ from s to d and the corresponding link flow allocation $Flow$;

Step 3 **if** $f_{\max} < |V_{TR}|$
  return $\emptyset$;
endif
$A \leftarrow \{(i,j),(j,i)\}$;
forall $e' = (x,y) \in E_2$ (where x corresponds to node v_k, y corresponds to node pair (v_h, v_l))
  if $Flow[e'] = 1$ **and** $k = h$
    $A \leftarrow A \cup \{(h,l)\}$;
  else
    $A \leftarrow A \cup \{(l,h)\}$;
  endif
endforall
return A;

Algorithm 6: AssignDOF(L, e)

transmitters and receivers corresponding to the link set $L \cup \{e\}$ respectively. We also define a node pair set $V_{TR} = \{(v_i, v_j) | v_i \in V_T, v_j \in V_R, v_i \in N_j, (v_i, v_j) \notin L \cup \{e\}\}$. Each of the first type of vertices in V' corresponds to a node in $V_T \cup V_R$. Each of the second type of vertices in V' corresponds to a node pair in V_{TR}. The corresponding vertex sets are denoted as V_1 and V_2 respectively. V' also includes an auxiliary source vertex s and destination vertex d. There is an edge from s to each vertex in V_1 with a capacity of $K-1$, where K is the number of DOFs at each node. In G', there are two edges going to each vertex in V_2 corresponding to a node pair (v_i, v_j), one from the vertex corresponding to v_i and another from the vertex corresponding to v_j, both of which has a capacity of 1. There is also an edge from each vertex in V_2 to d with a capacity of 1. The corresponding edge sets are denoted as E_1, E_2 and E_3 respectively. Hence, we have $V = V_1 \cup V_2 \cup \{s, d\}$ and $E = E_1 \cup E_2 \cup E_3$. Fig. 12 shows an example of the auxiliary graph G'.

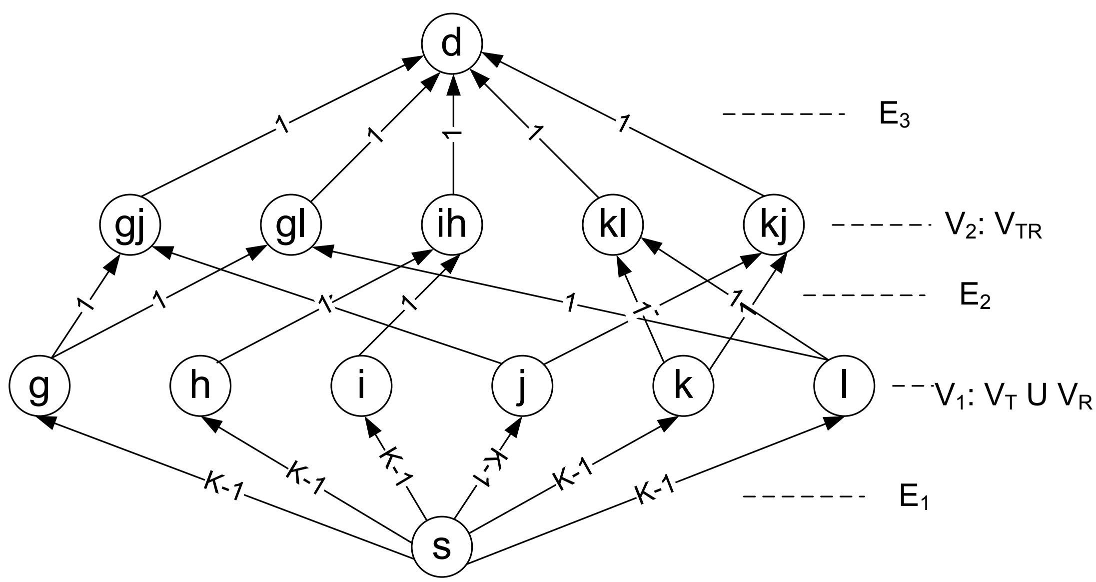


Figure 12: An example of auxiliary graph

In G', each vertex in E_2 actually corresponds to a potential secondary interference. We create two edges for such a vertex because a potential secondary interference can be eliminated by assigning a DOF at either the corresponding transmitter or the receiver. Every node has K DOFs, $K-1$ of which can be used to suppress secondary interference. Thus the capacity of each edge in E_1 is set to $K-1$. As mentioned before, an augmenting path based maximum flow algorithm such as the Ford-Fulkerson algorithm can always find a maximum flow whose corresponding link flows are all integers, if the capacity of each link is an integer. Therefore, if the Ford-Fulkerson algorithm can find a maximum flow of $|V_{TR}| = |E_3|$ in G', then there exists a feasible DOF assignment such that all potential secondary interference are cancelled. Otherwise, we know the given link e cannot be active concurrently with links in L.

We show that a link flow allocation from s to d in G' corresponds to a DOF assignment. Note that each edge $(u, f) \in E_2$ selected in the link flow allocation corresponds to assigning one DOF at the node u to eliminate the secondary interference f. Next, each edge in E_3 has capacity 1, so each vertex in F can contribute at most 1 to the flow. This corresponds to the fact that a secondary interference only needs to be eliminated once, either by the transmitter or by the receiver. Finally, each edge in E_1 has capacity $K-1$, so each vertex in $V_T \cup V_R$ can contribute at most $K-1$ to the flow. This corresponds to the number of DOFs at each node that can be used to eliminate secondary interferences.[9] In Step 2, we find the maximum flow in G', which corresponds to a DOF assignment to eliminate maximum number of secondary interferences. If this flow number is less than the number of secondary interferences among the links, it is impossible to eliminate all secondary interferences. Otherwise, the algorithm returns a DOF assignment A in Step 3, where each $(x, y) \in A$ means to assign a DOF of node x pointing at node y.

In Algorithm 6, Step 1 takes $O(|L|^2)$ time. In Step 2, G has $2|L|+|F|+2$ vertices and $2|L| + 3|F|$ edges. The Ford-Fulkerson algorithm in Step 3 takes $O(|E'|f_{\max})$ time. Since in G', $f_{\max} \leq |F|$, Step 3 takes $O((|L| + |F|)|F|)$ time. Step 4 loops for

[9] Our algorithm can be applied to networks where nodes have different number of DOFs.

$|f_{\max}|$ times and each loop takes $O(1)$ time. Overall, Algorithm 6 takes $O(|L|^2) + O(|L| + |F|) + O((|L| + |F|)|F|) + O(|F|) = O(|L|^2 + |F|^2)$ time.

In Algorithm 5, Step 1 takes $O(n^2)$ time to initiate $\mathbf{\Gamma}$. Step 2 sorts $e \in E$ in $O(n \log n)$ time, and for each $e \in E$ runs Algorithm 6 once, which takes $O(|L|^2 + |F|^2) = O(n^2)$ time. The Step 2 and 3 will execute for T^u times. Therefore, the overall time of the algorithm is $O(n^2) + T^u(O(n \log n + nO(n^2)) = O(n^3 T^u)$. However, in practice, the size of the set L of links that can be active in a minislot and the set F of secondary interferences are usually much smaller than $|E|$, the total number of links. Therefore, the algorithm has decent performance.

JOINT SCHEDULING AND CHANNEL ASSIGNMENT

In this chapter, we study a scheduling problem in OFDMA-based WRN with consideration for both diversity gains and spatial reuse. Similar to a closely related work [4], the objective is to keep the system stable, i.e., keep the length of every queue finite. Such a stable scheduling is also considered to achieve 100% or maximum throughput [41].

Problem Definition

We define the scheduling problem in the following.

Definition 3 *Given m links, K blocks, the queue length vector $\mathbf{Q}$, the interference matrix $\mathbf{I}$, and the data rate matrix $\mathbf{R}$. The scheduling problem seeks an interference-free block assignment that assigns a subset B_i of blocks to each link e_i such that the utility function $\sum_{i=1}^{m} q_i \min\{q_i, \sum_{k \in B_i} r_i^k\}$ is maximized.*

We choose the above objective function because it is known that if a scheduling algorithm can achieve the above objective in each frame or minislot, then it can keep the system stable, i.e., keep the length of each queue finite [4]. As mentioned before, such a stable scheduling algorithm is also considered to achieve maximum throughput [41].

Mixed Integer Linear Programming (MILP) Formulation and Solution

Next, we present an MILP formulation for the scheduling problem. The decision variables are described as follows.

1. $x_i^k = 1$ if block k is assigned to link e_i, $x_i^k = 0$ otherwise. (mK such variables)

2. y_i, the effective utility obtained by link e_i according to the assignment. (m such variables)

MILP:

$$\text{Maximize } \sum_{i=1}^{m} q_i y_i \tag{5}$$

subject to:

$$y_i \le q_i, \qquad i \in \{1, \dots, m\}; \tag{6}$$

$$y_i \le \sum_{k=1}^{K} r_i^k x_i^k, \qquad i \in \{1, \dots, m\}; \tag{7}$$

$$\sum_{j:I_{ij}=1} (x_j^k + x_i^k - 1) \le 0, \quad i \in \{1, \dots, m\}, k \in \{1, \dots, K\}; \tag{8}$$

$$x_i^k \in \{0, 1\}, \quad i \in \{1, \dots, m\}, k \in \{1, \dots, K\}. \tag{9}$$

In this formulation, constraints (6) and (7) make sure that $y_i = \min\{q_i, \sum_{k \in B_i} r_i^k\}$. Constraint (8) is the interference constraint, which ensures that if two links interfere with each other, they are not assigned the same block. The objective is to maximize the aforementioned utility function.

As mentioned before, a similar scheduling problem in OFDMA-based WiMAX single-hop networks has been shown to be NP-hard in [4]. It is a special case of the problem studied here since in a single-hop wireless network, all links interfere with each other and no special reuse is possible. Therefore, our problem is much harder.

The MILP formation of the problem can be solved exactly by an ILP solver [42]. However, since ILP is NP-hard, ILP solvers can only solve problems of small sizes in reasonable time. Therefore, non-exact faster algorithms are necessary to solve this problem in reality.

In this chapter, we will present two greedy algorithms to solve the scheduling problem defined in the last section, and show that they have constant approximation ratios. In addition, we will present 3 heuristics, namely, the MWIS algorithm, the sequential knapsack algorithm and the LP rounding algorithm.

The basic assignment unit is the link-block pair (i, k), which means to assign the block k to the link e_i. For the convenience of presentation, we call a link-block pair as a *unit* in the following descriptions of all algorithms. We also denote the set of the $m \times K$ link-block pairs (units) as L. The output of any algorithm is a block assignment matrix $\mathbf{X}$, in which $x_i^k = 1$ means the unit (i, k) is selected in the solution.

Simple Greedy Algorithm

The greedy algorithm is formally presented as Algorithm 7.

Step 1 $\mathbf{X} \leftarrow \mathbf{0}$;
$\mathbf{Q}' \leftarrow \mathbf{Q}$;

Step 2 $(i_{\text{best}}, k_{\text{best}}) \leftarrow \text{argmax}_{(i,k)}\, q_i \min\{q_i', r_i^k\}$;
if $q_{i_{\text{best}}} \min\{q'_{i_{\text{best}}}, r_{i_{\text{best}}}^{k_{\text{best}}}\} = 0$
return $\mathbf{X}$;

Step 3 $x_{i_{\text{best}}}^{k_{\text{best}}} \leftarrow 1$;
$q'_{i_{\text{best}}} \leftarrow \max\{0, q'_{i_{\text{best}}} - r_{i_{\text{best}}}^k\}$;
$L \leftarrow L \setminus \{(i_{\text{best}}, k_{\text{best}})\}$;
forall $j : I_{ij} = 1$ **do**
$L \leftarrow L \setminus \{(j, k_{\text{best}})\}$;
endforall
if $L = \emptyset$
return $\mathbf{X}$;

Step 4 **goto** Step 2;

Algorithm 7: Greedy

In Algorithm 7, q'_i is the remaining queue length of the link e_i after some blocks allocated to e_i. After initialization in Step 1, the greedy algorithm always selects a unit $(i_{\text{best}}, k_{\text{best}})$, i.e., assign block k_{best} to link $e_{i_{\text{best}}}$, such that the utility gain is maximum in Step 2. Step 3 updates the remaining queue length $q'_{i_{\text{best}}}$ and the block assignment matrix $\mathbf{X}$. The selected unit $(i_{\text{best}}, k_{\text{best}})$ is then removed from L. In addition, the algorithm ensures that those links interfering with the link $e_{i_{\text{best}}}$ are not assigned block k_{best} by removing the corresponding units from L. The algorithm terminates when no more utility gain can be achieved or L becomes empty, which means either all blocks have been assigned or all queues are satisfied.

Theorem 5 *For any network with an interference degree of $\delta \geq 1$, the simple greedy algorithm has an approximation ratio of $\frac{1}{1+\delta}$.*

Proof: Each unit (i, k) in L is evaluated in Step 2 of Algorithm 7 and is ranked by its marginal utility gain, defined as follows:

Let S be the set of blocks already allocated to e_i, then

$$gain(i, k, S) \equiv q_i \min(q_i, \sum_{k' \in S \cup \{k\}} r_i^{k'}) - q_i \min(q_i, \sum_{k' \in S} r_i^{k'}). \tag{10}$$

The unit $(i, k) \in L$ with the highest marginal utility gain is *allocated* and removed. Block k is assigned to e_i. And for all links e_j that interfere with e_i, each incompatible unit (j, k) is removed from L. We say that (j, k) was *blocked* by (i, k). Generally, all units are either allocated or blocked.[1]

Let X be the list of units selected by the greedy algorithm (in the order that they were selected) and $\overset{*}{X}$ be an optimal list of units (in some fixed order). We also

[1]In practice the algorithm stops allocating units when none have positive gain, which means all queues are 100% satisfied and the solution is optimum.

define the block lists $K_i = \{k : (i,k) \in X\}$ and $\overset{*}{K}_i = \{k : (i,k) \in \overset{*}{X}\}$.[2] For any block $k \in K_i$, let $K_{i,<k}$ be the blocks in K_i that precede k. Define $\overset{*}{K}_{i,<k}$ similarly. For each unit $(i,k) \in X$, let $g_i^k = gain(i,k,X_{i,<k})$.[3] Likewise, $\forall (i,k) \in \overset{*}{X}, \overset{*}{g}{}_i^k = gain(i,k,\overset{*}{X}_{i,<k})$. The value of the objective function (5) achieved by the greedy algorithm is $V = \sum_i \sum_{k \in X_i} g_i^k$ and the optimal value is $\overset{*}{V} = \sum_i \sum_{k \in \overset{*}{X}_i} \overset{*}{g}{}_i^k$.

For each link e_i, we classify $\overset{*}{K}_i$ into 4 disjoint sets. Let $K_i^{\mathrm{al}} = \{k \in \overset{*}{K}_i \cap K_i : \overset{*}{g}{}_i^k > g_i^k\}$, $K_i^{\mathrm{ag}} = \{k \in \overset{*}{K}_i \cap K_i : \overset{*}{g}{}_i^k \leq g_i^k\}$, $K_i^{\mathrm{bl}} = \{k \in \overset{*}{K}_i \setminus K_i : \overset{*}{g}{}_i^k > g_i^k\}$, and $K_i^{\mathrm{bg}} = \{k \in \overset{*}{K}_i \setminus K_i : \overset{*}{g}{}_i^k \leq g_i^k\}$.[4]

Then we classify the indices of all links into 3 disjoint sets.

Let $A = \{i : K_i^{\mathrm{al}} \cup K_i^{\mathrm{bl}} \neq \emptyset\}$. Note that if $i \in A$, there is a block k that provided or could have provided strictly less gain to e_i than it did in the optimal solution, which implies that the queue of e_i is fully satisfied and therefore $\sum_{k \in K_i} g_i^k = q_i^2 \geq \sum_{k \in \overset{*}{K}_i} \overset{*}{g}{}_i^k$.

Let $B = \{i : K_i^{\mathrm{al}} \cup K_i^{\mathrm{bl}} \cup K_i^{\mathrm{bg}} = \emptyset\}$. Therefore, $\forall i \in B, \overset{*}{K}_i = K_i^{\mathrm{ag}} \subseteq K_i$, and $\overset{*}{g}{}_i^k \leq g_i^k$.

Let $C = \{i : K_i^{\mathrm{al}} \cup K_i^{\mathrm{bl}} = \emptyset \wedge K_i^{\mathrm{bg}} \neq \emptyset\}$. Thus, $\forall i \in C, \overset{*}{K}_i = K_i^{\mathrm{ag}} \cup K_i^{\mathrm{bg}}$. And for each $k \in K_i^{\mathrm{bg}}$, (i,k) was blocked by some $(i',k) \in X$ with $\overset{*}{g}{}_i^k \leq g_i^k \leq g_{i'}^k$. We add these units (i',k) to $\mathcal{M}$, which is a multiset because some units in it may block multiple[5] units that were selected by $\overset{*}{X}$.

Observe that $A \cup B \cup C = \{1, \ldots, m\}$ and we have:

---

[2]With the same orderings as the unit lists

[3]The marginal utility gain of the unit (i,k) in the solution X

[4]al: allocated to a loss; ag: allocated to a gain; bl: blocked to a loss; bg: blocked to a gain

[5]At most δ, because $\forall (i,k) \in \mathcal{M}$, e_i interferes with at most δ non-interfering links.

$$
\begin{aligned}
\overset{*}{V} &= \sum_{i\in A}\sum_{k\in \overset{*}{K}_i} \overset{*}{g}{}_i^k + \sum_{i\in B}\sum_{k\in \overset{*}{K}_i} \overset{*}{g}{}_i^k + \sum_{i\in C}\sum_{k\in \overset{*}{K}_i} \overset{*}{g}{}_i^k \\
&\le \sum_{i\in A} q_i^2 + \sum_{i\in B}\sum_{k\in \overset{*}{K}_i} g_i^k + \sum_{i\in C}\sum_{k\in K_i^{\mathrm{ag}}} \overset{*}{g}{}_i^k + \sum_{i\in C}\sum_{k\in K_i^{\mathrm{bg}}} \overset{*}{g}{}_i^k \\
&\le \sum_{i\in A}\sum_{k\in K_i} g_i^k + \sum_{i\in B}\sum_{k\in K_i} g_i^k + \sum_{i\in C}\sum_{k\in K_i} g_i^k + \sum_{(i,k)\in\mathcal{M}} g_i^k \\
&\le V + \delta V \\
&= (1+\delta)V \qquad (11)
\end{aligned}
$$

Thus $V \ge \frac{1}{1+\delta}\overset{*}{V}$ as claimed. □

Combining Theorem 5 with Theorem 1 and 2 in Chapter 4, we have:

Corollary 1 *The greedy algorithm has an approximation ratio of* $\frac{1}{5}$ *for any 2-hop WRN and* $\frac{1}{15}$ *for any WiMAX network.*

The greedy algorithm takes $O(mK)$ time to compare all units and find one to schedule in Step 2. Step 2 dominates the loop from Step 2 to Step 4, which executes $|L|$ times. Therefore, its time complexity of the greedy algorithm is $O(mK|L|) = O(m^2K^2)$.

Weighted Degree Greedy Algorithm

The simple greedy algorithm always selects the unit with the most marginal utility gain even if this may cause the block not able to serve a set of links that would contribute more utility gain. To improve, we propose an algorithm that makes the greedy choice based on the ratio of the gain and the possible loss.

We first define the *contention graph* $G_C = (V_C, E_C)$, where each vertex in V_C corresponds to a link, and there exists an undirectional edge connecting two vertices if the corresponding links interfere with each other. The weight of each vertex (link) on block k is set to g_i^k, which is the marginal utility gain achieved by assigning block k to link e_i. We then define the *weighted interference degree* of link e_i in the contention graph G_C on block k as $d_w^k(i, G_C) \equiv \frac{\max_{E_i} \sum_{e_j \in E_i} g_j^k}{g_i^k}$, where E_i is any set of non-interfering links that interfere with link e_i. Usually, there are multiple non-interfering link sets for a link e_i. We select the set with the maximum total utility gain to calculate the weighted interference degree. Note that the weight and weighted interference degree are defined with the marginal utility gain, and therefore depended on the block set that are already assigned. The weighted degree greedy algorithm is formally presented as Algorithm 8.

Step 1 $\mathbf{X} \leftarrow \mathbf{0}$;
$\mathbf{Q}' \leftarrow \mathbf{Q}$;

Step 2 $(i_{\text{best}}, k_{\text{best}}) \leftarrow \text{argmin}_{(i,k)}\, d_w^k(i, G_C)$;
if $g_{i_{\text{best}}}^{k_{\text{best}}} = 0$
return $\mathbf{X}$;

Step 3 $x_{i_{\text{best}}}^{k_{\text{best}}} \leftarrow 1$;
$q'_{i_{\text{best}}} \leftarrow \max\{0, q'_{i_{\text{best}}} - r_{i_{\text{best}}}^k\}$;
$L \leftarrow L \setminus \{(i_{\text{best}}, k_{\text{best}})\}$;
forall $j : I_{i_{\text{best}},j} = 1$ **do**
$L \leftarrow L \setminus \{(j, k_{\text{best}})\}$;
endforall
if $L = \emptyset$
return $\mathbf{X}$;

Step 4 **goto** Step 2;

Algorithm 8: WeightedGreedy

Next we study Algorithm 8's performance. We define the *weighted inductiveness* of the graph G_C on block k as $\delta_w^k(G_C) \equiv \max_{H_C \subseteq G_C} \min_{e_i \in H} d_w^k(i, H_C)$. Here H_C is an arbitrary subgraph of G_C and may have different weights than G_C. The weighted inductiveness of a graph G_C on all blocks is defined as $\delta_w(G_C) \equiv \max_k \delta_w^k(G_C)$. The weighted inductiveness captures the mininum weighted degree of a subgraph, which corresponds to the unit that will be selected in the algorithm. It then find the maximum of such minimum weighted degrees over all subgraphs of G_C, which considers the worst case.

Theorem 6 *For a network with weighted inductiveness δ_w, the weighted degree greedy algorithm is $\frac{1}{1+\delta_w}$-approximate.*

Proof: The proof is similar to that of Theorem 5. The only difference is that for each $k \in K_i^{\text{bg}}$, (i, k) was blocked by some $(i', k) \in X$ with $\overset{*}{g_i^k} \leq g_i^k$. Notice that since i' corresponds to the vertex with the minimum weighted node degree, for all units (i, k) blocked by (i', k), $\sum_{(i,k)} g_i^k \leq d_w^k(i', G_C) g_{i'}^k \leq \delta_w g_{i'}^k$. This demonstrates that although we cannot guarantee that $g_i^k \leq g_{i'}^k$ for each i like we did for the simple greedy algorithm, we do guarantee that the summation of those marginal gains of non-interfering links is at most δ_w times the marginal gain of the link selected by the weighted greedy algorithm. Therefore, we have $\sum_{i \in C} \sum_{k \in \overset{*}{K}_i} \overset{*}{g_i^k} \leq \delta_w V$, instead of $\sum_{i \in C} \sum_{k \in \overset{*}{K}_i} \overset{*}{g_i^k} \leq \delta V$ in 11. Thus, $\overset{*}{V} \leq (1 + \delta_w) V$. $\square$

Theorem 7 *For any graph, $\delta_w \leq \delta$.*

Proof: For any subgraph $H_C \subseteq G_C$, we consider the link $e_i \in H_C$ with the maximum marginal gain g_i^k. There are at most δ non-interfering links that interfere with e_i, and each of them has a marginal gain of at most g_i^k. Therefore, $d_w^k(i, H_C) \leq \delta$, and $\delta_w \leq \delta$. $\square$

Combining Theorem 6 with Theorem 1, 2 and 7, we have:

Corollary 2 *The weighted degree greedy algorithm has an approximation ratio of* $\frac{1}{5}$ *for any 2-hop WRN and* $\frac{1}{15}$ *for any WiMAX network.*

Most of the time, $\delta_w < \delta$, and therefore the weighted degree algorithm has a better approximation ratio than the simple greedy algorithm. It should also have a better performance in real networks, which we will show in the following chapter.

In Step 2, the weighted degree of each link can be computed in polynomial time, since according to Theorem 1 and 2, the set of non-interfering neighbor links has constant size.[6] Hence all such sets can be enumerated in polynomial time. The other parts of the algorithm are the same as the simple greedy algorithm. Therefore, the entire algorithm is polynomial-time.

Maximum Weighted Independent Set (MWIS) Algorithm

Unlike those two greedy algorithms which select a unit at a time, in Algorithm 9, we consider blocks one by one. Essentially, a block can be allocated to a maximal set of non-interfering links to maximize the spacial reuse. The Step 1 initializes the variables, in which k denotes the block currently considering. Steps 2 and 3 determines which subset of non-interfering links the current block should be given to. The objective is to maximize the utility gain. We construct the contention graph to assist the computation. In the contention graph G_C, each vertex corresponds to a link and there is an undirectional edge connecting two vertices if the corresponding links interfere with each other. Moreover, the weight of each vertex (link) is set to $w_i \leftarrow q_i \min\{q'_i, r_i^k\}$, which is the utility gain achieved by assigning block k to link e_i.

[6]Bounded by 4 for 2-hop WRN and 14 for any WiMAX network

Step 1 $\mathbf{X} \leftarrow \mathbf{0}$;
$\mathbf{Q}' \leftarrow \mathbf{Q}$;
$k \leftarrow 1$;

Step 2 Construct a contention graph G_C;

Step 3 Find an MWIS E_{IS} in G_C;
forall $e_i \in E_{IS}$ **do**
$x_i^k \leftarrow 1$;
$q_i' \leftarrow \max\{0, q_i' - r_i^k\}$;
endforall

Step 4 **if** $k = K$
return $\mathbf{X}$;
$k \leftarrow k + 1$;
goto Step 2;

Algorithm 9: MWIS-Scheduling

In each step, the maximum utility gain can be achieved by finding an MWIS on the contention graph. However, the MWIS problem is a well-known NP-hard problem [43]. Therefore, we use Algorithm 10, the greedy approximation algorithm described in [43], to compute a maximal independent set E_{IS} in G_C (i.e., a subset of links). This algorithm repeatedly selects a vertex with minimum weighted vertex degree, puts it into the result independent set and removes it and all its neighbors from G, until G becomes empty. The weighted vertex degree of vertex v in G is defined as $d_w(v) = \frac{\sum_{v' \in N(v)} w_{v'}}{w_v}$, where N_v denotes the set of neighbors of vertex v in G_C. This algorithm has been shown of having an approximation ratio of $\frac{1}{1+\eta}$, where $\eta = \max_{v \in G_C} d_w(v)$ [43].

Conjecture 1 *If the algorithm used to solve the MWIS subproblem in Step* 3 *of Algorithm 9 has an approximation ratio of* $\frac{1}{\rho}$*, then Algorithm 9 has an approximation ratio of* $\frac{1}{1+\rho}$.

Step 1 $IS \leftarrow \emptyset$;

Step 2 **if** $G = \emptyset$
  return IS;

Step 3 $v \leftarrow \text{argmin}_{v \in V(G)} d_w(v, G)$;

Step 4 $IS \leftarrow IS \cup \{v\}$;

Step 5 Remove v and its neighbors from G;

Step 6 **goto** Step 2;

Algorithm 10: MWIS-Greedy(G)

The idea of proof is similar to that of Theorem 5. The only difference is that the multiplicity of any unit $(i, k) \in \mathcal{M}$ is at most ρ, which means the optimum solution may assign block k to serve another set of non-interfering links instead of the link e_i selected greedily with the minimum weighted degree, and the utility achieved by the optimum solution on those links is at most ρ times the utility achieved greedily.

We observe that the maximum weighted independent set subproblem can be solved exactly in polynomial time for any 2-hop network because the proof of Theorem 1 guarantees that the size of any independent set in such case is at most 4, and we can enumerate all of them in polynomial time. Therefore,

Corollary 3 *The MWIS algorithm has an approximation ratio of $\frac{1}{2}$ for any 2-hop relay network if Conjecture 1 is true.*

Note that we do not have a similar constant approximation ratio for the general h-hop relay networks because the proof of Theorem 2 limits the interference ratio, instead of the size of any independent set of the entire network.

Next we analyze the time complexity of Algorithm 9.

In Algorithm 10, it takes $O(m\Delta)$ time to find the vertex with minimum weighted degree in Step 3. m is the number of vertices in G, which equals the number of links in the network. Δ is the maximum node degree in G. Step 4 can be done in constant time. It takes $O(\Delta^3)$ time for Step 5 since there are $O(\Delta)$ neighbors of v and for each neighbor, we need to find its neighbors, which is $O(\Delta)$, and to remove a vertex from any vertex list in the adjacency list representation of G takes $O(\Delta)$ time. The loop from Step 2 to Step 6 will execute for $|IS|$ times. To sum up, the entire time complexity of Algorithm 10 is $O(\Delta^3|IS|) = O(m^2\Delta^2)$.

In Algorithm 9, the loop from Step 2 to Step 4 executes for K times and Step 3 dominates the running time. Therefore, its total time complexity is $O(m^2\Delta^2K)$.

Sequential Knapsack Algorithm

In this algorithm, we solve the problem from a new perspective. As pointed out in [4], the problem of finding a set of blocks to satisfy a link's demand is similar to the knapsack problem. Hence, we can solve our scheduling problem by solving a sequence of knapsack-like subproblems. After solving each subproblem, we make a *temporary* block assignment for a link. We call it temporary because a block k assigned to a link e_i may be reassigned to another link e_j later if the total utility would increase. The algorithm is described in detail as follows.

In Step 2, we construct a knapsack-like subproblem. Note that the utility loss is the total utility already achieved by assigning block k to some other links which interfere with link e_i. If we now decide to reassign block k to link e_i, we will lose those achieved utility. Step 3 solves the subproblem and updates the assignment matrix $\mathbf{X}$. We repeat Step 2 and 3 until all links have been examined. The knapsack-like

Step 1 Sort all links in the descending order of q_i;
$\mathbf{X} \leftarrow \mathbf{0}$;
$i \leftarrow 1$;

Step 2 Construct a knapsack-like problem with K blocks for link e_i (Details are given below).
For each block k, set the utility loss $\beta_k \leftarrow \sum_{j:I_{ij}=1} x_j^k q_j r_j^k$.
Set the remaining queue length $q_i' \leftarrow q_i$.

Step 3 Solve the knapsack-like problem and obtain a set B_i of blocks for link e_i;
forall $k \in B_i$ **do**
  $x_i^k \leftarrow 1$;
  forall $j : I_{ij} = 1$ **do**
    $x_j^k \leftarrow 0$;
  endforall
endforall

Step 4 **if** $i = m$
  return $\mathbf{X}$;
$i \leftarrow i + 1$;
goto Step 2;

Algorithm 11: Sequential-Knapsack

problem:

$$\text{Maximize } q_i z_i - \sum_{k=1}^{K} x_i^k \beta_k, \tag{12}$$

subject to:

$$x_i^k \in \{0,1\}, \quad \forall k \in \{1,\dots,K\}; \tag{13}$$

$$z_i \leq q_i; \tag{14}$$

$$z_i \leq \sum_{k=1}^{K} x_i^k r_i^k. \tag{15}$$

In this problem, z_i is the effective data rate achieved on link e_i by picking some blocks. We try to maximize the net utility gain in (12), which is the utility achieved on link e_i minus the utility loss from removing blocks previously assigned to other interfering links. Constraints (14) and (15) ensure that the effective data rate is the minimum of the queue length and the total data rate of all blocks selected. This problem is different from the classical knapsack problem because:

1. The total utility gain is not the simple summation of utility gains by each block selected. Because of the queue length constraint, utility gain acquired by some block may be partialy wasted. This reflects the characteristic of our original scheduling problem that the selected blocks for a link may provide more data rates than the length of the queue at the link.

2. Each block may have a utility loss, which is always β_k. This reflects the possible utility loss of reassigning a block.

We propose an algorithm to solve this knapsack-like problem in the following.

We define $U(k, q_i')$ as the maximum achievable utility using the first k blocks, when the remaining queue length is q_i'. Then we can derive this recursive formula:

$$U(k, q_i') = \begin{cases} 0, & \text{if } k = 0 \text{ or } q_i' \leq 0, \\ A, & \text{if } 0 < q_i' \leq \frac{\beta_k}{q_i}, \\ \max\{A, q_i q_i' - \beta_k\}, & \text{if } \frac{\beta_k}{q_i} < q_i' \leq r_i^k, \\ \max\{A, B\}, & \text{otherwise.} \end{cases} \tag{16}$$

where $A = U(k-1, q_i')$ is the maximum utility without picking block k, $B = q_i r_i^k - \beta_k + U(k-1, q_i' - r_i^k)$ is the maximum utility with block k picked and fully utilized.

In Equation (16), the first case is a trivial recursion terminator. The second case says that, if the remaining queue length q_i' is so small that the utility gain of picking block k cannot even offset its utility loss, then block k should not be considered. In the third case, there is some profit by picking block k. If we pick it, then the remaining queue is fully served and no more block needs to be picked. This is because all items have already been sorted by their profit/weight ratio. Therefore picking any other item would not result in more benefit. Instead, it will only introduce additional overhead. So we compare the utility of picking block k with A and return the better one. The last case is the general one, where we decide whether to pick block k by comparing A and B.

Equation (16) is implemented using dynamic programming with memoization.[7] Each time a subproblem $U(k, q_i')$ is solved, the set of blocks selected is returned together with the utility value.

The knapsack-like sub-problem is solved in Algorithm 12 and 13. Algorithm 12 provides the interface and call Algorithm 13 to do the actual computing.

---

[7] using either search tree or hash table

Step 1 Sort **P**, **Q**, **W** by $\frac{p_i}{w_i}$ ascending;

Step 2 **return** DoKnapsack($|\mathbf{P}|, W$);

Algorithm 12: Knapsack($\mathbf{P}, \mathbf{Q}, \mathbf{W}, W$)

Step 1 **if** $n = 0$ **or** $W = 0$
    return $\emptyset$;

Step 2 **if** the solution to input (n, W) has been computed
    return memoized solution;

Step 3 $A \leftarrow$ DoKnapsack($n - 1, W$);

Step 4 **if** $W \leq \frac{w_n}{p_n} q_n$
    return A;

Step 5 **if** $W \leq w_n$
    return $\max\{A, \{n\}\}$;

Step 6 $B \leftarrow \{n\} \cup$ DoKnapsack($n - 1, W - w_n$);
  Memoize $(n, W) \Rightarrow \max\{A, B\}$;
  return $\max\{A, B\}$;

Algorithm 13: DoKnapsack(n, W)

Algorithm 12 and 13 return a set of item indices which are selected into the knapsack. The max($\cdot$) operation in Algorithm 13 calculates the utility of each set in comparison and returns the set with better utility.

The knapsack-like problem can be solved in $O(Kq_i)$ time and is therefore pseudo-polynomial. Note that in the knapsack-like problem, n corresponds to a block and W corresponds to the queue length of a link in the original problem. It is solved for m times in Algorithm 11. Therefore, the total time complexity of our sequential knapsack algorithm is $\sum_i O(Kq_i) = O(KQ)$, where Q is the sum of the queue lengths of all links.

Our sequential knapsack algorithm shares the same idea of solving a sequence of knapsack-like problem and allocating resource in a link-by-link fashion with the *MaxWeight-Alg4* in [10]. However, [10] considered only 1-hop networks without spacial reuse. Their problem is a special case of our problem in multi-hop networks with spacial reuse. Therefore, our resulted knapsack-like problem is also more difficult. And we propose a different dynamic programming approach to solve it.

Linear Programming (LP) Rounding Algorithm

LP rounding is a common approach to solve ILP/MILP problems. The critical part of LP rounding is the rounding scheme, in which we decide which variable(s) should be rounded to 1. We find out that a trivial rounding scheme usually results in poor performance for our problem. This is because the maximum effective data rate achievable for a link is bounded by q_i, its queue length. Therefore, it is very likely that the rate summation of all the blocks allocated to a link outnumbers the queue length. In this case, some of the channel capacity is wasted and variables x_i^k are not tightly fixed in the LP solution. In other words, it is possible to increase some

x_i^k and decrease another $x_i^{k'}$ while keeping the LP solution's feasibility and utility. However, such randomness in the LP solution may result in different MILP solutions and often causes a low quality MILP solution. In Algorithm 14, we propose a novel rounding scheme. Our idea is to figure out the maximum possible value of each variable in the optimum LP solution to eliminate the aforementioned randomness. Unlike other rounding schemes, we consider two factors for a rounding decision: the value of each variable in the LP solution and the maximum possible utility gain given by this variable.

Step 1 Sort all the units in L in the descending order of $q_i r_i^k$;

Step 2 Solve the LP relaxation $\mathcal{P}'$ of the MILP problem $\mathcal{P}$ defined in Section 7;

Step 3 **forall** $(i, k) \in L$ **do**
  if $x_i^k = 1$ in the solution of $\mathcal{P}'$
    Round(x_i^k);
  else
    break;
  endif
endforall

Step 4 **forall** $(i, k) \in L$ **do**
  Solve a new LP problem $\mathcal{P}''$ which is the same as $\mathcal{P}'$ except that its objective is to maximize x_i^k and it has one more constraint fixing the utility function (5)'s value to that given by the solution of $\mathcal{P}'$;
  if $x_i^k = 1$ in the solution of $\mathcal{P}''$
    Round(x_i^k);
  endif
endforall

Step 5 Select (i, k) with the maximum value of x_i^k in the solution of $\mathcal{P}''$;
Round(x_i^k);

Step 6 **if** $L = \emptyset$
  return X;
goto Step 2;

Algorithm 14: LP-Rounding

Our LP rounding algorithm starts by sorting units according to their maximum achievable utility gains. Next, we relax the original MILP problem to an LP problem, which can then be solved by any existing LP solving algorithm [44]. In Step 3, we find those link-channel pairs that are not only selected by the LP solution but also appear on the top of L. They should certainly be selected. So we call the Round($\cdot$) subroutine which rounds the corresponding variables x_i^k to 1 and all variables $x_j^k : I_{ij} = 1$ to 0, removes all the corresponding link-block pairs from L, and updates $\mathcal{P}'$. The Round($\cdot$) subroutine guarantees that the interference constraints in the original MILP are not violated during the rounding process. Step 3 terminates when we find the channel-block pair not selected by the LP solution with the largest utility gain. Notice that failure to select this unit in the LP solution may just be due to the randomness. We eliminate such randomness in Step 4 by finding the maximum possible x_i^k value for each unit that has not been selected so far under the constraint that the utility function value is conserved, i.e., finding their *potentials*. This can be done by solving a series of LPs. In effect, Step 4 looks for more variables and confirm their assignments in the LP solution. So far our algorithm has not rounded any variable from a fractional value to 1 yet. Consequently, the utility achieved by the LP still remains optimum. After Step 4, if none of the remaining undecided variable could be rounded to 1 without reducing the utility, we select a link-block pair with the largest value of x_i^k and round the corresponding variable to 1 in Step 5. It is possible to select multiple non-conflicting variables in Step 5 and round all of them to 1 at once. This will speed up the entire rounding algorithm at the cost of obtaining a final solution with likely inferior quality. In our implementation, we round only one variable at a time. The algorithm terminates when all variables are decided.

Next, we analyze the time complexity of Algorithm 14 in terms of the total number of LPs solved. Step 2 solves 1 LP. In Step 4, at most $|L|$ LPs need to be solved. The loop from Step 2 to Step 6 will round at lease one variable per loop. Therefore the loop will execute for at most $|\mathbf{X}|$ times, where $|\mathbf{X}|$ is the size of the solution. The total number of LPs solved is at most $O(|L||\mathbf{X}|) = O(nK|\mathbf{X}|)$, which is polynomial to the input size. And each LP can be solved in polynomial time. Therefore, the LP rounding algorithm is a polynomial time algorithm. In the worst case, we may need to solve many LPs. However, the algorithm is time efficient in practice because it can often round many variables in each loop, and once a variable has been rounded to 1 (a unit is selected), all variables corresponding to the interfering units will be rounded to 0. We may not even need to solve all the LPs. Because all of them have just slight modification to the original LP and should be efficiently solvable by modifying the original LP solver algorithm.

SIMULATIONS AND ANALYSIS

Routing Tree Construction, Joint Scheduling and DOF Assignment

The performance of the proposed algorithms for routing tree construction (in Chapter 5) and joint scheduling and DOF assignment (in Chapter 6) were evaluated via extensive simulations. The simulations were implemented using Microsoft Visual C++ and LEDA [45], a commercial graph library.

Scenario Settings

In the simulations, the nodes were uniformly deployed within a 4×8 km^2 rectangular region, with a BS at the top-left corner. The number of nodes (network size) varied from 25 to 150, with step size of 25. According to [1], the number of minislots per frame was set to 1024. The uplink and downlink demands of an SS were uniformly distributed in $[5, 10]$ and $[10, 20]$ respectively. The transmission and interference range was set to 1 km and 3 km respectively.

Since our work is the first to address WiMAX scheduling with smart antennas, we compare our scheduling algorithms with the first-fit algorithm and a trivial solution. The first-fit algorithm is a typical greedy algorithm, which tries to pack as many links with unsatisfied bandwidth demands as possible in the first minislot in a top-down fashion without violating the interference constraints, and then repeats this procedure for the next minislot until all minislots are used. The trivial solution is mentioned in the WiMAX standard [1], which does not allow spatial reuse (i.e., only one link is active in each minislot). In terms of routing, we compared the trees constructed by our algorithm with those by the Minimum Spanning Tree algorithm (MST) and the Breadth First Search algorithm (BFS). The end-to-end throughput, the minimum satisfaction ratio and the well-known Jain's fairness index [46][1] for the uplink traffic are used as the performance metrics.

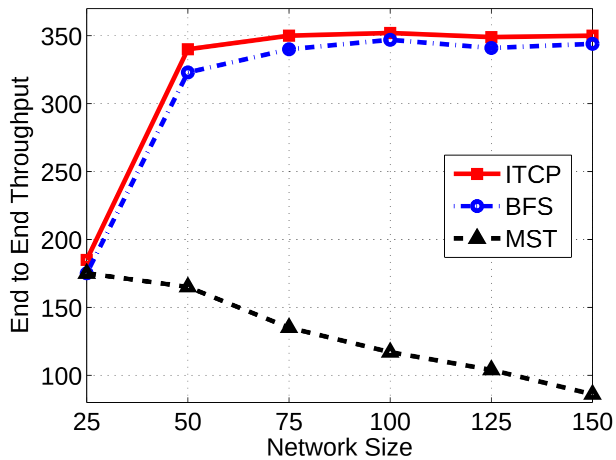
Simulation Results

In the first scenario, we compared different tree construction algorithms and scheduled the transmissions using our scheduling algorithm proposed for the general case. The corresponding results are presented in Fig. 13. In scenarios 2 and 3, we evaluated the performance of different scheduling algorithms for the special and general cases respectively. Our algorithm for solving the ITCP is always used to construct the routing tree. The corresponding results are presented in Figs. 14 and 15 respectively. In scenario 4, we evaluated the performance of different complete solutions (scheduling + routing). Refer to Fig. 16 for the results. For scenarios 1, 3, and 4, the number of DOFs at each node was set to $K = 3$. Each result presented in the figures is the average over 100 simulation runs. In each run, a network is randomly generated and used for all algorithms. In these figures, "USP" stands for our uplink scheduling algorithm for the special and general cases and "ITCP" represents our tree construction algorithm.

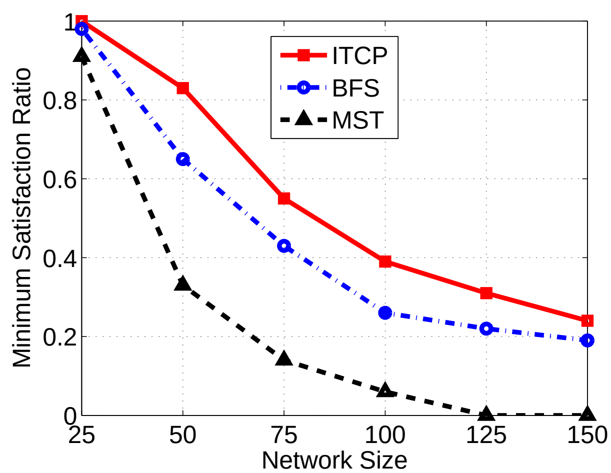
We make the following observations from Figs. 13–17:

1. As shown in Fig. 13, compared to the BFS and MST algorithms, our tree construction algorithm improves the minimum satisfaction ratio by 21% and 120%, the fairness index by 15% and 10%, and the end-to-end throughput by 3% and 140%, respectively. Essentially, more links in an end-to-end path (larger tree height) will normally lead to worse performance because it is more likely that allocating enough resources to an end-to-end path may fail. According

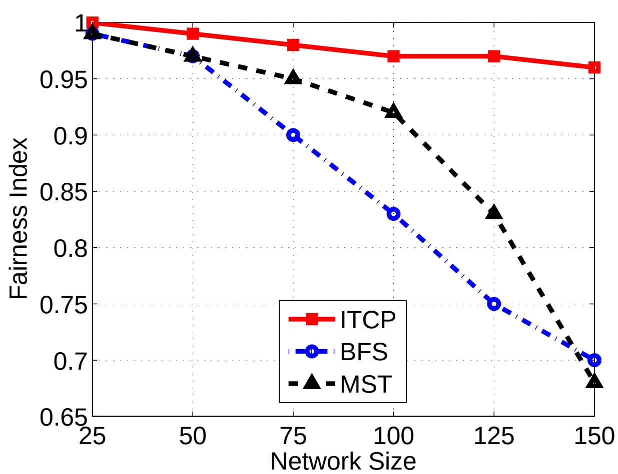
[1] fairness $= \frac{(\sum_i x_i)^2}{n \sum_i x_i^2}$, where x_i is the satisfaction ratio of a node.



(a) End-to-end throughput

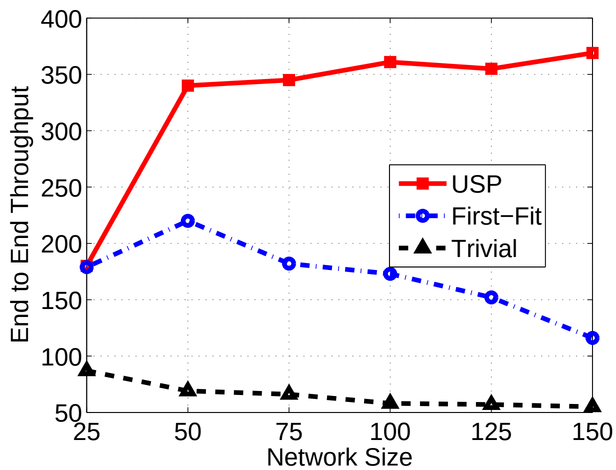


(b) Minimum satisfaction ratio

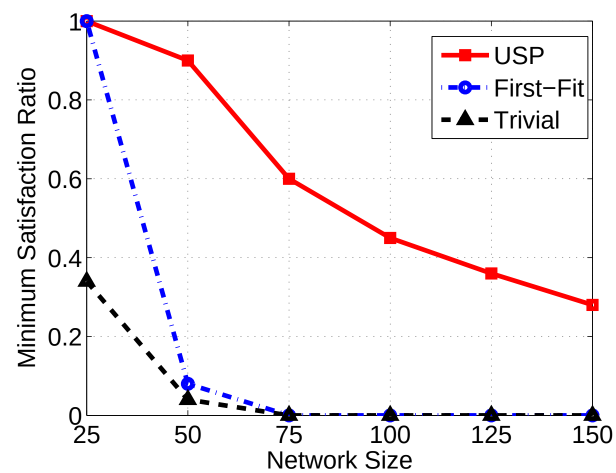


(c) Fairness index

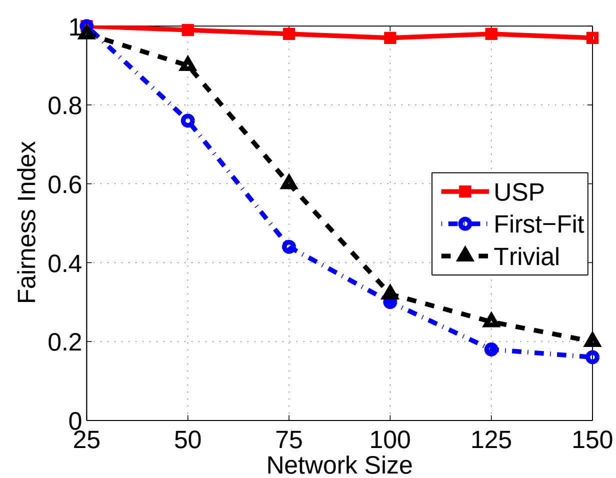
Figure 13: The tree construction algorithms



(a) End-to-end throughput

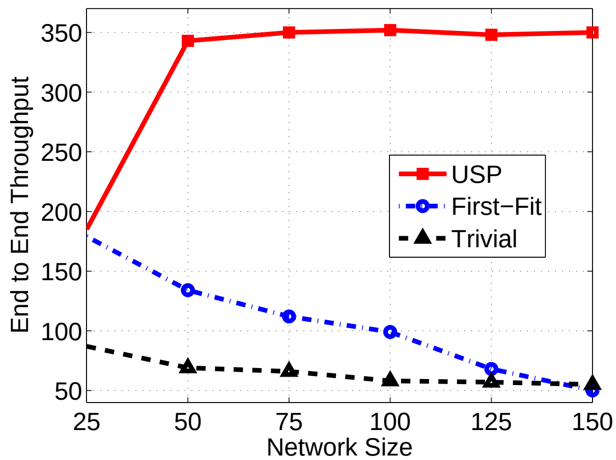


(b) Minimum satisfaction ratio

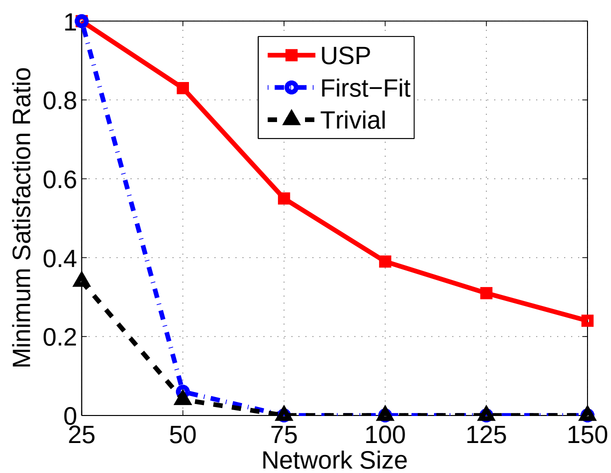


(c) Fairness index

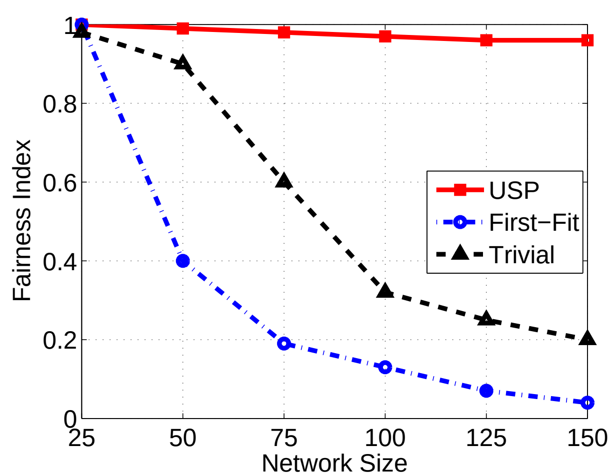
Figure 14: The scheduling algorithms for the special case



(a) End-to-end throughput

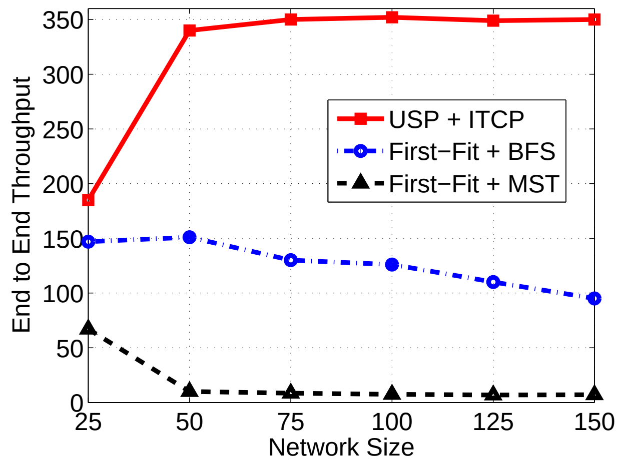


(b) Minimum satisfaction ratio

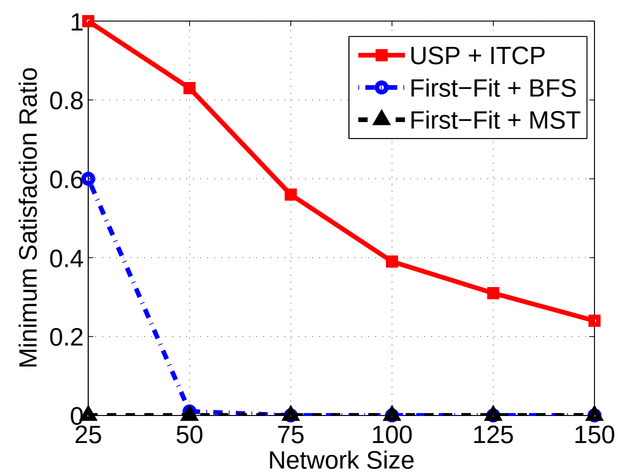


(c) Fairness index

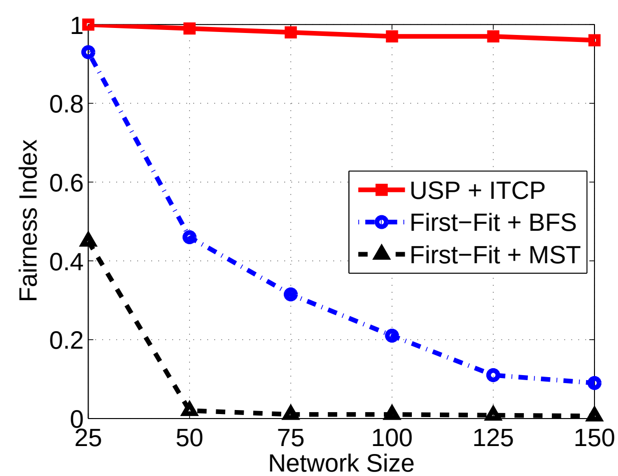
Figure 15: The scheduling algorithms for the general case



(a) End-to-end throughput

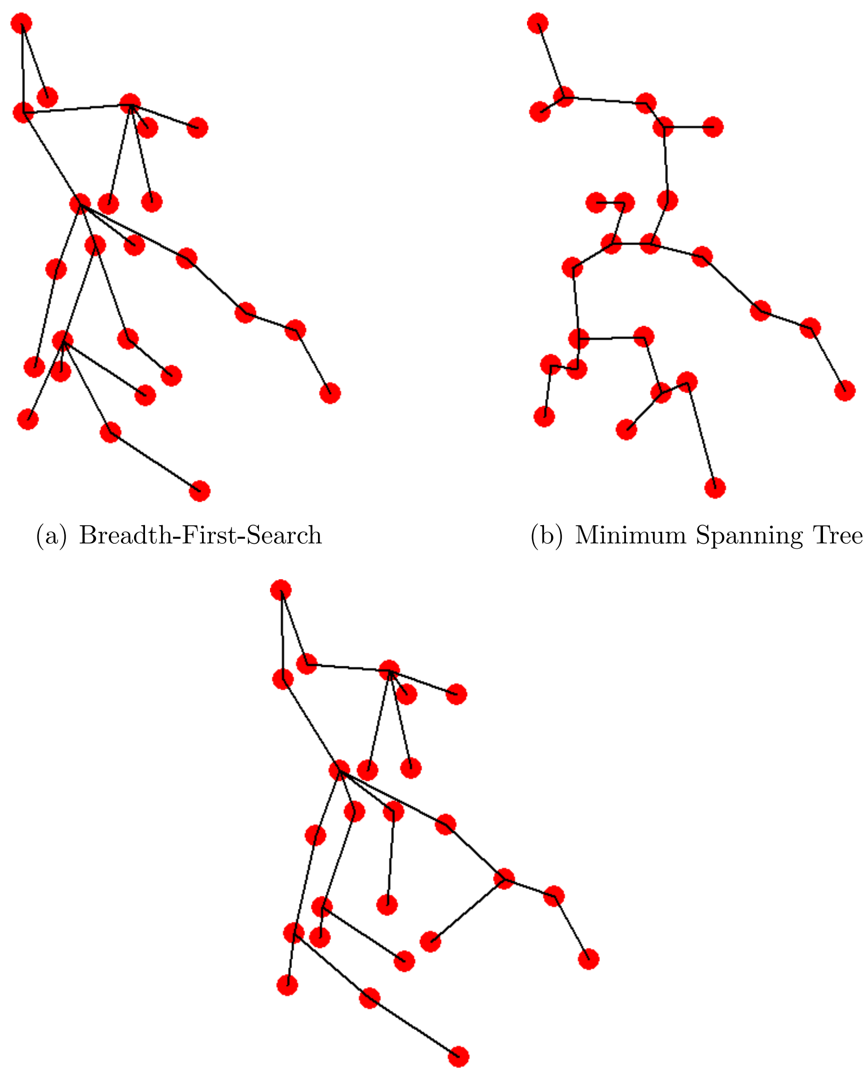


(b) Minimum satisfaction ratio



(c) Fairness index

Figure 16: The complete solutions



(a) Breadth-First-Search

(b) Minimum Spanning Tree

(c) Our ITCP algorithm

Figure 17: Example trees contructed by different algorithms

to our observations, an MST tree usually has a larger height than the tree constructed by our routing algorithm. A BFS tree has smaller height than an MST tree. However, the BFS trees are usually unbalanced, i.e., a particular node may have a relatively large number of descendants, which is obviously a negative factor for achieving good performance. Check Fig. 17 for a typical result of trees constructed by different algorithms.

2. Our scheduling algorithms always perform the best in both the special and the general cases. Specifically, compared to the first-fit algorithm, our scheduling algorithm (for the general case) can significantly improve the end-to-end throughput by 213%, the minimum satisfaction ratio by 220%, the fairness index by 200% on average. Moveover, no matter how large the network is, the fairness indices given by our scheduling algorithms are always very close to 1.0, which indicates that our algorithms can achieve a fair bandwidth allocation. As expected, the trivial algorithm performs very poorly in terms of both throughput and fairness, since it does not take advantage of spacial reuse. The first-fit algorithm performs closely to our algorithm when the network size is small and almost all requests can be satisfied. As the network size grows, the first-fit algorithm acquires a higher end-to-end throughput, but with a lower fairness index and a very low minimum satisfaction ratio which means some of the nodes get very little or even no bandwidth allocated. As network size keeps growing and the network becomes saturated, the first-fit algorithm even has throughput decreased. This is because in such cases, first-fit allocates most minislots to links at highest layers, while links at lower layers almost do not get any minislot. Therefore, lots of minislots are wasted at high-layer links because no traffic can be relayed from low-layer nodes. Our algorithm can

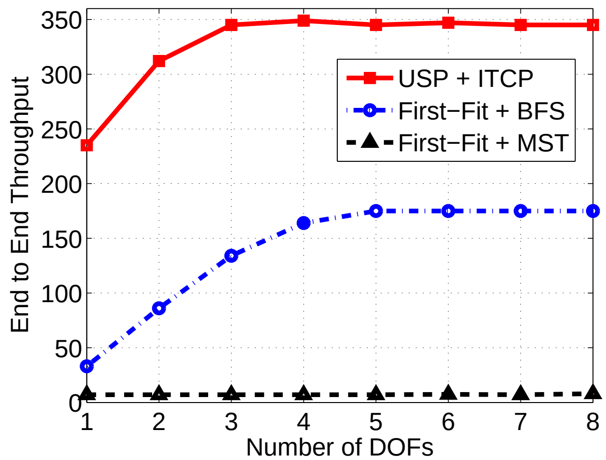
find a balanced tree with relatively small height. The tree constructed by BFS has fewer levels than MST. So most traffic flows have fewer hops to BS, and therefore BFS has better throughput than MST. However, BFS has the worst fairness index because it is likely for a node to have a very large subtree and result in an imbalanced tree.

3. Not surprisingly, the complete solution using our scheduling and routing algorithm significantly outperforms all other solutions. Specifically, compared to the first-fit+BFS solution, our solution achieves an average improvement of 154% on the end-to-end throughput, 446% on the minimum satisfaction ratio, and 177% on the fairness index.

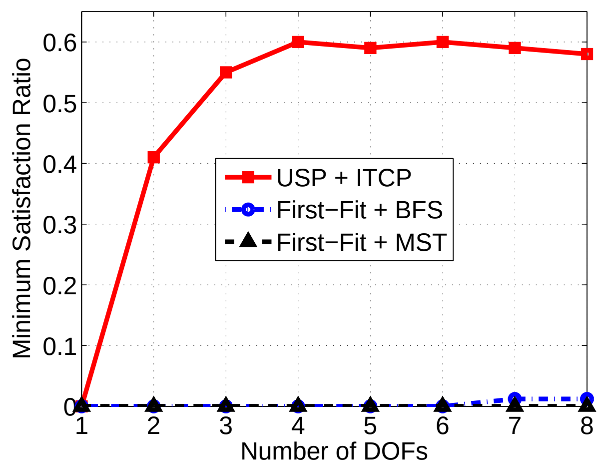
4. A denser (larger) network has heavier traffic demands, and is supposed to result in higher throughput. Therefore, we can see from Figs. 14–16 that the end-to-end throughput given by our scheduling and routing algorithms always increases with the network size. However, more SSs introduce stronger interference, which will hold back the throughput improvement on the contrary. Therefore, the throughput given by algorithms without carefully addressing the impacts of interference such as the first-fit algorithm may even decrease with the network size. In addition, the minimum satisfaction ratio and the fairness index always decrease with the network size because it is more difficult to achieve high fairness in a larger network.

In the second scenario, we fix the network size to 75 and examine the performances of three complete solutions with secondary interferences and different number of DOFs from 1 to 8. The results are presented in Fig. 18.

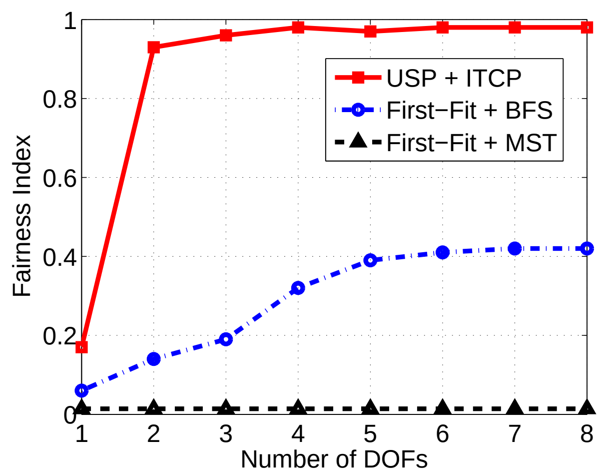
We make the following observations from Fig. 18:



(a) End to end throughput



(b) Minimum satisfaction ratio



(c) Fairness index

Figure 18: Complete solution performance with different DOFs

1. All three solutions have better performance when more DOFs are available. However, no matter how many DOFs are available in a node, our complete solution always has the best performance.

2. Because our solution conducts optimal DOF assignment, it can use the fewest DOFs to reach the optimal performance. Therefore, it enables the usage of DAAs with fewer DOFs to acquire interference free and reduces the network infrastructure cost.

Joint Scheduling and Channel Assignment

In this section, we evaluate the performance of the proposed algorithms for the joint scheduling and channel assignment in OFDMA-based WRN (in Chapter 7) by simulation. ILOG CPLEX 10.0 [42] is used to solve all the LP and MILP problems.

Scenario Settings

In our simulations, we set the transmission range to 1km and the interference range to 2km. For scenarios regarding general WiMAX networks, we uniformly place all nodes in a 5km $\times$ 5km square region. For scenarios regarding 2-hop WRN, we uniformly place all nodes in a circle with radius of 2km. The BS is always placed at the center of the region. We use the Breadth First Search (BFS) algorithm to construct the routing tree rooted at the BS.

We calculate path loss for all links using the model proposed in [47]. The effective propagation exponent calculated from the model is 3.3. In order to reflect the heterogeneous channel fading, we let the path loss fluctuate around the calculated mean value according to the Rayleigh distribution. According to the WiMAX standard, there are 7 adaptive modulation schemes: BPSK$\frac{1}{2}$, QPSK$\frac{1}{2}$, QPSK$\frac{3}{4}$, 16QAM$\frac{1}{2}$, 16QAM$\frac{3}{4}$, 64QAM$\frac{2}{3}$, and 64QAM$\frac{3}{4}$ with 0.5, 1, 1.5, 2, 3, 4, and 4.5 bits per symbol respectively. In the network, the symbol rate is fixed and therefore we normalize the data rates to $1, 2, 3, 4, 6, 8, 9$. Since we assume that all nodes use the same transmission power level, the data rate on a link is computed by applying a stair function on the path loss, resulting in one of the 7 possible data rates.

Traffic demands (queue lengths) are randomly generated according to binormial distribution. The mean value is fixed for all nodes in one scenario but varies for different scenarios.

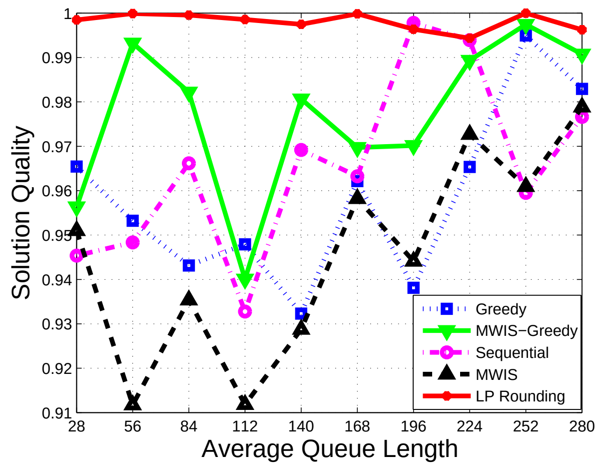
Simulation Results

The simulation results are shown in Fig. 19 and 20. In the scenarios shown in Fig. 19, we fix the network size to 30 and vary the mean queue length. In each scenario, we compare the solution quality of the 5 proposed algorithms against the optimum solution calculated by the ILP solver. The solution quality is measured as the percentage of ulility achieved by ILP. We make the following observations:

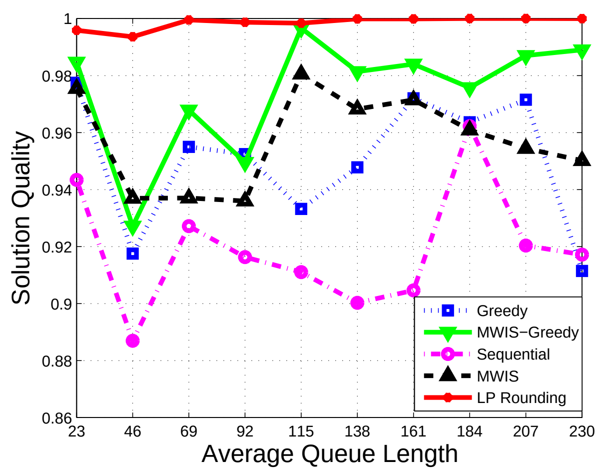
1. All the algorithms have decent performance. They achieve at least 91% of the optimum solution's utility. Their performance are also stable despite of different levels of network load.[2]

2. The LP rounding algorithm performs best. For networks with 6 channels and 4 minislots, the LP rounding algorithm outperforms the simple greedy algorithm by 6.82%, the weighted degree greedy algorithm by 2.06%, the sequential knapsack algorithm by 11.62%, and the MWIS algorithm by 5.52%, on average. For networks with 8 channels and 16 minislots, the LP rounding algorithm outperforms the simple greedy algorithm by 5.15%, the weighted degree greedy

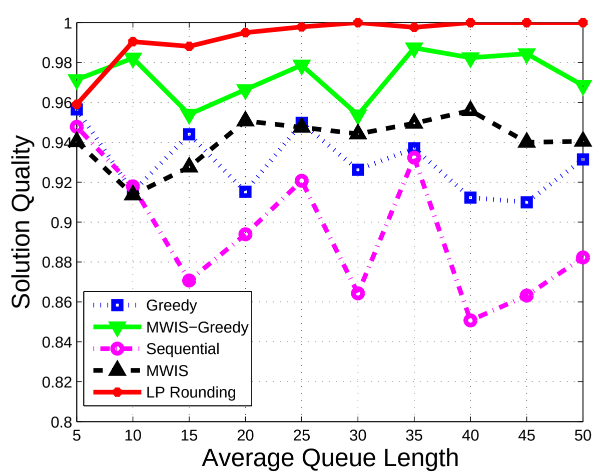
[2] The longer the mean queue length, the heavier the network is loaded.



(a) 2-hop with 30 nodes and 128 blocks

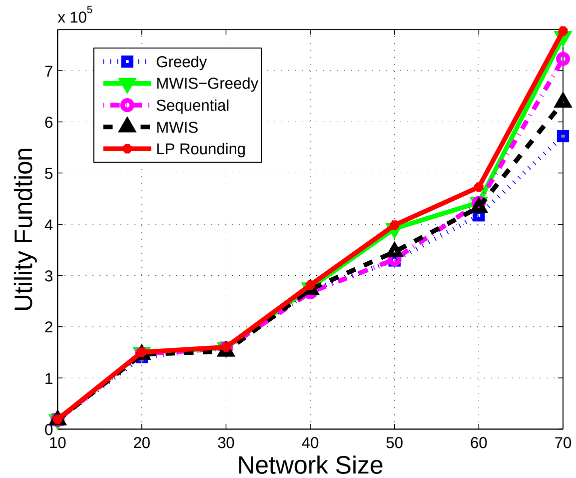


(b) h-hop with 30 nodes and 128 blocks

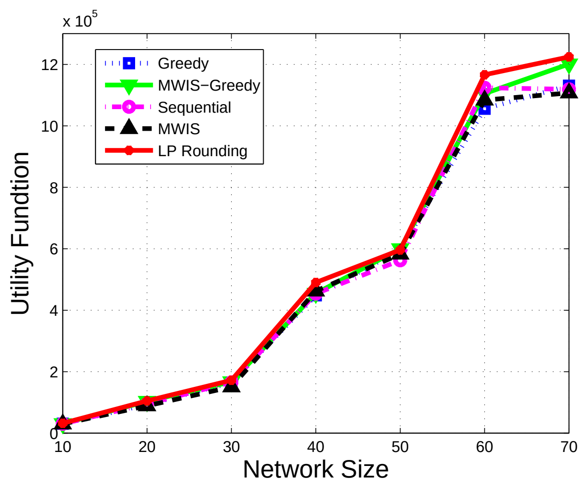


(c) h-hop with 30 nodes and 24 blocks

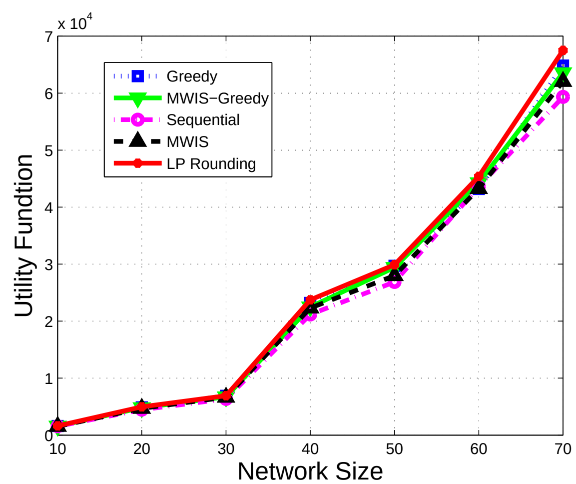
Figure 19: Quality against queue length



(a) 2-hop with 8 channels and 16 minislots



(b) h-hop with 8 channels and 16 minislots



(c) h-hop with 6 channels and 4 minislots

Figure 20: Utility against network size

algorithm by 2.53%, the sequential knapsack algorithm by 8.72%, and the MWIS algorithm by 4.36%, on average. For 2-hop networks with 8 channels and 16 minislots, the LP rounding algorithm outperforms the simple greedy algorithm by 4.16%, the weighted degree greedy algorithm by 2.19%, the sequential knapsack algorithm by 3.87%, and the MWIS algorithm by 5.64%, on average.

On average, the LP rounding algorithm outperforms the simple greedy algorithm by 5.4%, the weighted degree greedy algorithm by 2.3%, the sequential knapsack algorithm by 8.1%, and the MWIS algorithm by 5.2%. In most cases, the LP rounding algorithm can find a solution that is within 1% of utility difference to the optimum. It can often find the actual optimum solution. Therefore, this polynomial time algorithm is a good candidate when the network size is large and the execution time of the ILP solver becomes impractical.

3. Except when compared against the LP rounding algorithm, the greedy algorithms provide comparable performances. The greedy algorithms also have proved approximate ratios and thus can guarantee the performance in the worst case. Moreover, the greedy algorithms are easy to implement and take the least computation time, and are therefore suitable when the computation resource is limited. In most cases, the weighted degree greedy algorithm performs better than the simple greedy algorithm at the cost of longer computation time.

In scenarios shown in Fig. 20, we fix the mean queue length and vary the network size. In each scenario, we compare the solution quality of the 5 proposed algorithms. We do not show the result acquired by ILP solver for these scenarios because it would become impractical to use ILP solver for large networks. And for those small size networks that ILP solver can achieve optimum solutions, the utilities are very close to that found by the LP rounding algorithm. The solution quality is measured as the ulility value. We make the following observations:

1. The utility increases with the network size. This is because a larger network has more links, and therefore better chance to find links with higher data rates to assign blocks to.

2. Networks with 8 channels and 16 minislots (Fig. 20(b)) have higher utilities than networks with 6 channels and 4 minislots (Fig. 20(c)). This is obvious as there are more network resources.

3. With same number of channels and minislots, h-hop networks (Fig. 20(b)) have higher utilities than 2-hop networks (Fig. 20(a)). This is because in a 2-hop network, all nodes are close to each other. Therefore the interference level is high and the spatial reuse is very limited.

4. Similar as in the previous scenarios, the LP rounding algorithm performs the best and all other algorithms have comparable performances. For networks with 6 channels and 4 minislots, the LP rounding algorithm outperforms the simple greedy algorithm by 6.94%, the weighted degree greedy algorithm by 2.30%, the sequential knapsack algorithm by 9.47%, and the MWIS algorithm by 6.12%, on average. For networks with 8 channels and 16 minislots, the LP rounding algorithm outperforms the simple greedy algorithm by 6.99%, the weighted degree greedy algorithm by 3.31%, the sequential knapsack algorithm by 6.66%, and the MWIS algorithm by 9.29%, on average. For 2-hop networks with 8 channels and 16 minislots, the LP rounding algorithm outperforms the simple greedy algorithm by 12.09%, the weighted degree greedy algorithm by 2.69%, the sequential knapsack algorithm by 7.48%, and the MWIS algorithm

by 9.18%, on average. On average, the LP rounding algorithm outperforms the simple greedy algorithm by 8.7%, the weighted degree greedy algorithm by 2.8%, the sequential knapsack algorithm by 7.9%, and the MWIS algorithm by 8.2%. For large networks, the MWIS algorithm has slightly better performance than the greedy algorithm and the sequential knapsack algoirthm.

CONCLUSIONS AND FUTURE WORK

Conclusions

In this dissertation, we study resource allocation problems in WRN using technologies such as OFDMA and smart antennas.

We study the interference degree in WiMAX networks and prove a tight bound of 4 in 2-hop WRN, and a loose bound of 14 in general WiMAX networks.

We formally define the Interference-aware Tree Construction Problem (ITCP) for routing and present a polynomial-time algorithm to solve it. It has been shown that the trees constructed by our algorithm outperform the well-known MST and BFS trees by simulations. We present a polynomial-time optimal algorithm for a special case of the joint scheduling and DOF assignment problem as well as an effective heuristic algorithm for the general case. Our simulation results show that compared with other solutions such as the first-fit+BFS solution, our interference aware routing and scheduling scheme can improve throughput by 154% and fairness index by 177% on average.

The scheduling problem in OFDMA-based WRN with consideration for multi-user diversity, channel diversity and spacial reuse is generally very hard to solve. An ILP solver can solve it exactly, but only for small problem instances. For large networks, we propose several heuristic algorithms. Based on interference degree theory, we show that both the simple greedy algorithm and our proposed weighted-degree based greedy algorithm have approximation ratio of $\frac{1}{5}$ for the 2-hop case and $\frac{1}{15}$ for the general case. Extensive simulations have shown that the LP rounding algorithm performs best and always provides close-to-optimum solution. The performance of the greedy algorithms are comparable to that of the other algorithms.

Future Work

WRN is a very complicated system. Therefore, there are numerous research possibilities on this topic. Based on our achieved results in this work, we suggest the following related topics for future research:

- Study the tight bound of the interference degree in general WiMAX networks.
- Study the interference degree in real networks and simulated networks.
- Consider the end-to-end throughput and delay for the joint scheduling problems in WRN.
- Consider scheduling with guarantees for WiMAX QoS service levels.
- Study the joint scheduling, channel assignment, and DOF assignment problem. This would be much harder than the scheduling with channel assignment problem and the scheduling with DOF assignment problem we considered in this work.

REFERENCES CITED

[1] IEEE 802.16 Working Group. IEEE standard for local and metropolitan area networks part 16: Air interface for fixed broadband wireless access systems. *IEEE Std 802.16—2004 (Revision of IEEE Std 802.16—2001)*, pages 0_1–857, 2004.

[2] IEEE 802.16 Working Group. IEEE 802.16 working group, part 16: Air interface for fixed and mobile broadband wireless access systems—multihop relay specification. *IEEE Standard*, 2007.

[3] IEEE 802.16 Working Group. IEEE 802.16 working group, part 16: Air interface for fixed and mobile broadband wireless access systems amendment 2: Physical and medium access control layers for combined fixed and mobile operation in licensed bands. *IEEE Standard*, 2005.

[4] Matthew Andrews, Lisa Zhang. Scheduling algorithms for multi-carrier wireless data systems. *MobiCom'07: the 13th Annual ACM International Conference on Mobile Computing and Networking*, pages 3–14, New York, NY, USA, 2007. ACM. `doi:http://doi.acm.org/10.1145/1287853.1287856`.

[5] Karthikeyan Sundaresan, Weizhao Wang, Stephan Eidenbenz. Algorithmic aspects of communication in ad-hoc networks with smart antennas. *MobiHoc'06: the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 298–309, Florence, Italy, May 22–25 2006. `doi:http://doi.acm.org/10.1145/1132905.1132938`.

[6] D. Kim, A. Ganz. Fair and efficient multihop scheduling algorithm for IEEE 802.16 BWA systems. *BroadNets'05: the 2nd International Conference on Broadband Networks*, Volume 2, pages 833–839, Boston, MA, USA, October 3–7 2005. `doi:10.1109/ICBN.2005.1589693`.

[7] Harish Shetiya, Vinod Sharma. Algorithms for routing and centralized scheduling to provide QoS in IEEE 802.16 mesh networks. *WMuNeP'05: the 1st ACM Workshop on Wireless Multimedia Networking and Performance Modeling*, pages 140–149, Montréal, Quebec, Canada, October 13 2005. `doi:http://doi.acm.org/10.1145/1089737.1089760`.

[8] Min Cao, V. Raghunathan, P. R. Kumar. A tractable algorithm for fair and efficient uplink scheduling of multi-hop WiMAX mesh networks. *WiMesh'06: the 2nd IEEE Workshop on Wireless Mesh Networks*, pages 93–100, Reston, VA, USA, September 25 2006. `doi:10.1109/WIMESH.2006.288622`.

[9] C. Cicconetti, I. F. Akyildiz, L. Lenzini. Bandwidth balancing in multi-channel IEEE 802.16 wireless mesh networks. *InfoCom'07: the 26th IEEE International Conference on Computer Communications*, pages 2108–2116, Anchorage, AK, USA, May 6–12 2007. `doi:10.1109/INFCOM.2007.274`.

[10] Matthew Andrews, Lisa Zhang. Creating templates to achieve low delay in multi-carrier frame-based wireless data systems. *InfoCom'08: the 27th IEEE International Conference on Computer Communications*, pages 861–869, Phoenix, AZ, USA, April 13–18 2008.

[11] Debalina Ghosh, Ashima Gupta, Prasant Mohapatra. Scheduling in multihop WiMAX networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 12(2):1–11, April 2008.

[12] S. Ramanathan. A unified framework and algorithm for channel assignment in wireless networks. *Wireless Networks*, 5(2):81–94, 1999. doi:http://dx.doi.org/10.1023/A:1019126406181.

[13] Weizhao Wang, Yu Wang, Xiang-Yang Li, Wen-Zhan Song, Ophir Frieder. Efficient interference-aware TDMA link scheduling for static wireless networks. *MobiCom'06: the 12th Annual International Conference on Mobile Computing and Networking*, pages 262–273, Los Angeles, CA, USA, September 24–29 2006. doi:http://doi.acm.org/10.1145/1161089.1161119.

[14] Liqun Fu, Zhigang Cao, Pingyi Fan. Spatial reuse in IEEE 802.16 based wireless mesh networks. *ISCIT'05: IEEE International Symposium on Communications and Information Technology*, Volume 2, pages 1358–1361, Beijing, China, October 12–14 2005. doi:10.1109/ISCIT.2005.1567120.

[15] Jian Tao, Fuqiang Liu, Zhihui Zeng, Zhangxi Lin. Throughput enhancement in WiMAX mesh networks using concurrent transmission. *WCNM'05: IEEE International Conference on Wireless Communications, Networking and Mobile Computing*, Volume 2, pages 871–874, Wuhan, Hubei, China, September 23–26 2005. doi:10.1109/WCNM.2005.1544189.

[16] Hung-Yu Wei, S. Ganguly, R. Izmailov, Z. J. Haas. Interference-aware IEEE 802.16 WiMAX mesh networks. *VTC'05-Spring: IEEE 61st Semiannual Vehicle Technology Conference*, Volume 5, pages 3102–3106, Stockholm, Sweden, May 30–June 1 2005. doi:10.1109/VETECS.2005.1543918.

[17] Karthikeyan Sundaresan, Sampath Rangarajan. On exploiting diversity and spatial reuse in relay-enabled wireless networks. *MobiHoc'08: the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 13–22, New York, NY, USA, 2008. doi:http://doi.acm.org/10.1145/1374618.1374622.

[18] S. Deb, V. Mhatre, V. Ramaiyan. WiMAX relay networks: opportunistic scheduling to exploit multiuser diversity and frequency selectivity. *MobiCom'08: the 14th Annual International Conference on Mobile Computing and Networking*, pages 163–174, 2008.

[19] Karthikeyan Sundaresan, Sampath Rangarajan. Efficient algorithms for leveraging spatial reuse in OFDMA relay networks. *InfoCom'09: the 28th IEEE International Conference on Computer Communications*, pages 1539–1547, Rio de Janeiro, Brazil, April 19–25 2009.

[20] Xiaojun Lin, Ness B. Shroff. Joint rate control and scheduling in multihop wireless networks. *CDC'04: the 43rd IEEE Conference on Desicion and Control*, Volume 2, pages 1484–1489, Atlantis, Paradise Island, Bahamas, December 14–17 2004.

[21] Özgür Oyman. OFDM^{2}A: A centralized resource allocation policy for cellular multi-hop networks. *ACSSC'06: the 40th Asilomar Conference on Signals, Systems and Computers*, pages 656–660, Pacific Grove, CA, USA, October 29–November 1 2006. `doi:10.1109/ACSSC.2006.354830`.

[22] Romit Roy Choudhury, Xue Yang, Ram Ramanathan, Nitin H. Vaidya. Using directional antennas for medium access control in ad hoc networks. *MobiCom'02: the 8th Annual International Conference on Mobile Computing and Networking*, pages 59–70, Atlanta, Georgia, USA, September 23–28 2002. `doi:http://doi.acm.org/10.1145/570645.570653`.

[23] Thanasis Korakis, Gentian Jakllari, Leandros Tassiulas. A MAC protocol for full exploitation of directional antennas in ad-hoc wireless networks. *MobiHoc'03: the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 98–107, Annapolis, MD, USA, June 1–3 2003. `doi:http://doi.acm.org/10.1145/778415.778428`.

[24] Yajing Xu, Shen Wan, Jian Tang, Richard S. Wolff. Interference aware routing and scheduling in WiMAX mesh networks with smart antennas. *SeCon'09: 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, June 2009.

[25] K. Sundaresan, R. Sivakumar, M. A. Ingram, Tae-Young Chang. Medium access control in ad hoc networks with MIMO links: optimization considerations and algorithms. *IEEE Transaction on Mobile Computing*, 3(4):350–365, October–December 2004. `doi:10.1109/TMC.2004.42`.

[26] Brendan Mumey, Jian Tang, Timothey Hahn. Joint stream control and scheduling in multihop wireless networks with MIMO links. *ICC'08: IEEE International Conference on Communications*, pages 2921–2925, Beijing, China, May 2008. `doi:10.1109/ICC.2008.550`.

[27] Karthikeyan Sundaresan, Raghupathy Sivakumar. A unified MAC layer framework for ad-hoc networks with smart antennas. *MobiHoc'04: the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages

244–255, Roppongi Hills, Tokyo, Japan, May 24–26 2004. `doi:http://doi.acm.org/10.1145/989459.989490`.

[28] Ming Hu, Junshan Zhang. MIMO ad hoc networks with spatial diversity: medium access control, saturation throughput, and optimal hop distance. *CDC'04: the 43rd IEEE Conference on Decision and Control*, Volume 3, pages 3301–3306, Atlantis, Paradise Island, Bahamas, December 14–17 2004.

[29] R. Bhatia, Li Li. Throughput optimization of wireless mesh networks with MIMO links. *InfoCom'07: the 26th IEEE International Conference on Computer Communications*, pages 2326–2330, Anchorage, AK, USA, May 6–12 2007. `doi:10.1109/INFCOM.2007.274`.

[30] Yih-Hao Lin, Tara Javidi, Rene L. Cruz, Laurence B. Milstein. Distributed link scheduling, power control and routing for multi-hop wireless MIMO networks. *ACSSC'06: the 40th Asilomar Conference on Signals, Systems and Computer*, pages 122–126, Pacific Grove, CA, USA, October 29–November 1 2006. `doi:10.1109/ACSSC.2006.356597`.

[31] Romit Roy Choudhury, Nitin H. Vaidya. Deafness: a MAC problem in ad hoc networks when using directional antennas. *ICNP'04: the 12th IEEE International Conference on Network Protocols*, pages 283–292, October 2004. `doi:10.1109/ICNP.2004.1348118`.

[32] Supratim Deb, Karan Mangla, K. V. M. Naidu. Fast and distributed computation of schedules in wireless networks. *InfoCom'08: the 27th IEEE International Conference on Computer Communications*, pages 1121–1129, Phoenix, AZ, USA, April 13–18 2008.

[33] Hari Balakrishnan, Christopher L. Barrett, V. S. Anil Kumar, Madhav V. Marathe, Shripad Thite. The distance-2 matching problem and its relationship to the MAC-layer capacity of ad hoc wireless networks. *IEEE Journal on Selected Areas in Communications*, (6):1069–1079, August 2004.

[34] Yung Yi, Mung Chiang. Wireless scheduling algorithms with $o(1)$ overhead for m-hop interference model. *ICC'08: IEEE International Conference on Communications*, pages 3105–3109, Beijing, China, May 2008. `doi:10.1109/ICC.2008.584`.

[35] Gaurav Sharma, Ravi R. Mazumdar, Ness B. Shroff. On the complexity of scheduling in wireless networks. *MobiCom'06: the 12th Annual International Conference on Mobile Computing and Networking*, pages 227–238, New York, NY, USA, 2006. ACM. `doi:http://doi.acm.org/10.1145/1161089.1161116`.

[36] Prasanna Chaporkar, Koushik Kar, Xiang Luo, Saswati Sarkar. Throughput and fairness guarantees through maximal scheduling in wireless networks. *IEEE Transactions on Information Theory*, 54(2):572–594, February 2008.

[37] C. Joo, X. Lin, N. B. Shroff. Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks. *InfoCom'08: the 27th IEEE International Conference on Computer Communications*, pages 1103–1111, Phoenix, AZ, USA, April 13–18 2008.

[38] X. Lin, S. Rasool. A distributed joint channel-assignment, scheduling and routing algorithm for multi-channel ad hoc wireless networks. *InfoCom'07: the 26th IEEE International Conference on Computer Communications*, pages 1118–1126, Anchorage, AK, USA, May 6–12 2007. `doi:10.1109/INFCOM.2007.274`.

[39] S. Kravitz. Packing cylinders into cylindrical containers. *Mathematics Magazine*, 40:65–71, 1967.

[40] Kamal Jain, Jitendra Padhye, Venkata N. Padmanabhan, Lili Qiu. Impact of interference on multi-hop wireless network performance. *Wireless Networks*, 11(4):471–487, 2005. `doi:http://dx.doi.org/10.1007/s11276-005-1769-9`.

[41] Andrew Brzezinski, Gil Zussman, Eytan Modiano. Enabling distributed throughput maximization in wireless mesh networks: a partitioning approach. *MobiCom'06: the 12th Annual International Conference on Mobile Computing and Networking*, pages 26–37, New York, NY, USA, 2006. `doi:http://doi.acm.org/10.1145/1161089.1161094`.

[42] ILOG Software Inc. CPLEX 10.1.

[43] A. Kako, T. Ono, T. Hirata, M. M. Halldórsson. Approximation algorithms for the weighted independent set problem in sparse graphs. *Discrete Applied Mathematics*, 157(4):617–626, 2009.

[44] M. S. Bazaraa, J. J. Jarvis, H. D. Sherali. *Linear Programming and Network Flows (3rd edition)*. John Wiley and Sons, 2005.

[45] Algorithmic Solutions. Library of efficient data types and algorithms (LEDA).

[46] R. Jain, D. M. Chiu, W. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems, 1984.

[47] H. Carneiro, J. Oliveira, A. Rodrigues, P. Sebastiao. Software planning tool for WiMAX networks. *the 6th Conference on Telecommunications*, 2007.

APPENDICES

APPENDIX A

GLOSSARY

802.11 — IEEE 802.11 is a set of standards for wireless local area network (WLAN) computer communication in the 5 GHz and 2.4 GHz public spectrum bands.

802.16 — The IEEE WiMAX standard set.

802.16j — An amendment to the IEEE 802.16 standard, focused on multihop relay specification.

BS (Base Station) — A radio transceiver that serves as the hub of the WiMAX network and the gateway to the external network, such as the Internet. In centralized scheme, BS is in charge of allocating the WiMAX network resource and coordinate the communication of other SSs.

diversity gain — The performance improvement of communnication resulted by using different time, frequency, path, etc to transmit the same message.

DOF (Degrees of Freedom) — The number of beams/nulls that a smart antenna can provide, usually equal to the number of elements in the array.

effective satisfaction ratio — The minimum satisfaction ratio of all descendants of a node.

interference degree — The maximum number of non-interfering links that interfere with a link.

the knapsack problem — One of the famous NP-complete problems. Given a set of items, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less than a given limit and the total value is as large as possible.

LP (Linear Programming) — One type of mathematical optimization problems in which all the objectives and constraints are linear to the variables.

MILP (Mixed Integer Linear Programming) — An LP with some variables limited to integer and other variables real numbers.

MIMO (Multiple Input Multiple Output) antenna — A technology to use multiple antennas at both the transmitter and receiver to improve the communication performance.

MWIS (Maximum Weighted Independend Set problem) — One of the famous NP-complete problems. To find an independent set in a graph with the maximum vertex weights of all vertices in the independent set. It is one form of smart antenna technology.

OFDM (Orthogonal Frequency-Division Multiplexing) — A frequency-division multiplexing scheme that uses a large number of closely-spaced orthogonal sub-carriers to carry data. Each sub-carrier is modulated with a conventional modulation scheme (such as quadrature amplitude modulation or phase-shift keying) at a low symbol rate.

OFDMA (Orthogonal Frequency Division Multiplexing Access) — Based on OFDM, OFDMA groups OFDM sub-carriers into subchannels and assign them to multiple users.

primary interference — When two unicast communication links are incident to each other (share a common node), they cannot be active simultaneously using half-duplex transceivers. This is defined as *primary interference* between these two links.

satisfaction ratio — The ratio of the bandwidth allocated to a node to the bandwidth request of the node.

secondary interference — When two links do not have primary interference in between, but they cannot be active simultaneously because at least one link would have Signal-to-Interference-and-Noise Ratio (SINR) dropped below the threshold, it is said to have *secondary interference* between these two links.

smart antenna — Also known as adaptive array antennas, a type of antenna which can adapt its transceiving to the environment, forming beams/nulls toward directions to enhance signal strength and supress interference.

SS (Subscriber Station) — A radio transceiver that serves as the interface to WiMAX network for a client/subscriber. The client/subscriber may be a device or a local area network. SS may communicate directly to the BS or use other SSs as relay in a multihop fashion.

TDMA (Time Division Multiple Access) — A method for multiple users to share a channel. The signal is divided to different time slots. Each user will only access the channel in his/her own time slots.

WiMAX — An acronym for *Worldwide Interoperability for Microwave Access*, is described by the WiMAX forum as “a standards-based technology enabling the delivery of last mile wireless broadband access as an alternative to cable and DSL”. It has been standardized as IEEE 802.16.

WRN (WiMAX Relay Network) — Defined in [2] to extend the WiMAX standard by introducing RSs (Relay Stations) to relay traffic between the BS and SSs.

APPENDIX B

NOTATIONS

All the major notations in this work are summarized in Table 1.

Table 1: Notations

$\mathbf{A}^u/\mathbf{A}^d$	The aggregated uplink/downlink bandwidth allocation vector
$\mathbf{B}^u/\mathbf{B}^d$	The uplink/downlink bandwidth allocation vector
$G = (V, E)$	The communication graph
$H[v]$	The layer of node v in the routing tree
$\mathbf{I}$	The interference matrix
I_{ij}	An entry in $\mathbf{I}$, $I_{ij} = 1$ if link e_i interferes with link e_j; $I_{ij} = 0$, otherwise. Let $I_{ii} = 0$.
$I^p(v)/I^s(v)$	The primary/secondary interference value of node v
$I^s(e)$	The secondary interference value of link e
$I^b(h)$	The secondary interference bound of layer h
K	The number of DOFs at each node
L	The set of $m \times K$ link-block pairs
n/m	The number of nodes/links in G
N_i	The set of nodes which can potentially interfere with v_i
p_i	The index of the parent node of v_i
q_i	The queue length of link e_i at the beginning of a frame
$\mathbf{Q}$	The queue length vector, $\mathbf{Q} = [q_1, \ldots, q_i, \ldots, q_m]$
$\mathbf{Q}^u/\mathbf{Q}^d$	The uplink/downlink bandwidth demand vector
$\mathbf{R}$	The data rate matrix
r_i^k	An entry in $\mathbf{R}$, the data rate of link e_i that can be supported by block k
R_T/R_I	The transmission/interference range
$T/T^u/T^d$	The number of minislots in a frame/uplink subframe/downlink subframe
$\mathbf{S}^u/\mathbf{S}^d$	The uplink/downlink satisfaction ratio vector
V_h/E_h	The set of nodes/links in layer h
$\mathbf{X}$	The block assignment matrix
x_i^k	An entry in $\mathbf{X}$, $x_i^k = 1$ if link e_i is assigned block k; $x_i^k = 0$, otherwise.
δ	The interference degree of the network
$\mathbf{\Gamma}$	The scheduling matrix
$\mathbf{\Lambda}$	The DOF assignment matrix

APPENDIX C

DESIGN OF SIMULATION PROGRAMS

WiMAX DAA scheduler

This program is implemented using Microsoft Visual C++ 6.0 and the LEDA library.

The program put all input constants in head_define.h. Outputs are directed to text files under the "Result" directory. LEDA library is used to show the trees generated by each tree construction algorithm.

main.cpp is the entry point of the program. It drives the entire simulation process using the Tree class, collects simulation data, and writes to the output files.

Tree.h and tree.cpp defines and implements all the data structures and algorithms in the program.

WiMAX OFDMA scheduler

Multiple programs are designed and implemented to generate scenarios and simulate for all the algorithms proposed in Chapter 7. All programs are implemented using Microsoft Visual C++ 2003 with the BOOST library.[1] The WiMAX_Scheduling program also uses CPLEX library to solve MILP and LP.

GenScenario

This program reads several parameters from the console input, randomly generates topology of a few general h-hop WiMAX relay networks, and outputs the results to "Network.n.txt", where "n" is the network size.

[1] http://www.boost.org

Functionalities: Nodes in the generated network are guaranteed to be connected in the network. In case some of the randomly generated points are disconnected, they will be discarded and regenerated.

After positions of all nodes are generated, the topology of the network is fixed and computed. The interferece map is computed as well.

Finally, path loss of each communication link is computed using Raleigh fading model. Then we calculate the data rate according to the path loss.

Design: Datarate.h and Datarate.cpp define and implement the Rayleigh random fading model. Path loss in dB and datarate in integer numbers can be calculated by the distance of the link.

Gen2Hops.cpp is the main module of this program. It randomly generates 2-hop network topologies with different number of nodes. It calls the Datarate module to get datarates for all links in the generated network.

Gen2Hops

This program is similar to the GenScenario program introduced in the last subsection, except that it generates 2-hop network topologies in a circle.

WiMAX_Scheduling

This program reads network topology from files generated by GenScenario.exe or Gen2Hops.exe, randomly generates traffic for each network, and uses various algorithms to solve the scheduling problem for each network. Outputs are output*.txt, which contains the utility and execution time for each algorithm, and lp_round*.txt, which contains detailed execution information about the LP rounding algorithm.

Design: The struct Problem (defined in Problem.h) and the class Solution (defined in Solution.h) describe the problem and solution to each algorithm. Each algorithm is a class inherited from the Scheduler interface. These algorithms are implemented:

1. Greedy: the simple greedy algorithm in Chapter 7, always looks for a ¡user, channel, symbol¿ combination that can provide the best utility up to the point.

2. AdvGreedy: greedy algorithm based on weighted node degree, instead of individual node weight used by the simple greedy algorithm.

3. AdvGreedy2: the weighted degree greedy algorithm in Chapter 7, similar to AdvGreedy, but uses weighted node inductiveness instead of weighted node degree.

4. ByChannel: MWIS algorithm in Chapter 7. This algorithm finds a maximal weighted independent set to schedule for each channel in each symbol. The algorithm to find the MWIS can be configured and is currently set to use a greedy heuristic based on weighted node degree.

5. BySymbol: this algorithm utilizes the MILP solver to find a maximum weighted independent set for each symbol, taking both channels and users into account. The solution found would be optimum if the system has only one symbol in a scheduling period. Generally, its solution is sub-optimum.

6. ByUser: the sequential knapsack-like algorithm in Chapter 7. Allocates resources in a user-by-user fashion.

7. MILP: uses the CPLEX MILP solver to find the optimum solution. Only works for small scenarios.

8. LPRounding1: LP rounding algorithm in Chapter 7. Slightly improved from LPRounding.

Other files:

- Cliques.h and Cliques.cpp implement the algorithm to calculate all maximal cliques/independent sets in a graph, which is used by many other algorithms in the program.
- MWIS.h defines the interface to solve the maximum weighted independent set problem.
- MWIS_Greedy.h and MWIS_Greedy.cpp implements the weighted node degree based greedy heuristic to solve MWIS.
- MWIS_ILP.h and MWIS_ILP.cpp utilizes the MILP solver to solve MWIS.
- Network.h, Network.cpp, Scenario.h, Scenario.cpp defines classes for problem definition. They are used by class Problem.