DISTANCE-AWARE RELAY PLACEMENT AND SCHEDULING IN WIRELESS NETWORKS

A Thesis
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Shi Bai

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

June 2011

Fargo, North Dakota

North Dakota State University
Graduate School

Title

DISTANCE – AWARE RELAY PLACEMENT

AND SCHEDULING IN WIRELESS NETWORKS

By

SHI BAI

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

ABSTRACT

Bai, Shi, M.S., Department of Computer Science, College of Science and Mathematics, North Dakota State University, June 2011. Distance-Aware Relay Placement and Scheduling in Wireless Networks. Major Professors: Dr. Weiyi Zhang and Dr. Jun Kong.

The WiMAX technology and cognitive radio have been active topics in wireless networks. A WiMAX mesh network is able to provide larger wireless coverage, higher network capacity and Non-Line-Of-Sight (NLOS) communications. Cognitive radios enable dynamic spectrum access over a large frequency range. These characteristics make WiMAX mesh networks and cognitive radio networks able to provide users with low-cost, high-speed and long-range wireless communications, as well as better Quality of Service. However, there are still several challenges and problems to be solved in this area, such as relay station placement problems and scheduling problems. In this thesis, I studied a distance-aware relay placement problem and max-min fair scheduling problem in WiMAX mesh networks. To solve these problems, approximation algorithms and heuristic algorithms are proposed. Theoretical analysis and simulation results are provided to evaluate the solutions. I also studied a scheduling problem adopting the idea of cognitive radio technique in wireless networks over water. Two heuristics are presented to solve this unique problem. I provide the numerical results to justify the performance and efficiency of our proposed scheduling algorithms.

ACKNOWLEDGEMENTS

I would like to thank my advisor Dr. Weiyi Zhang; without his inspiration, help and support, my research work at NDSU could not have been accomplished. It is really a glorious experience that I have been working with Dr. Weiyi Zhang. I also would like to thank my committee members, Dr. Jun Kong, Dr. Tariq King, and Dr. Chao You for their time and help on my thesis and defense. I also thank my friends Yang Liu and Chao Liu. I really appreciate their help on my research work and life.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

The WiMAX technology (IEEE 802.16) is the fourth generation standard for low-cost, high-speed and long-range wireless communications for a large variety of civilian and military applications. IEEE 802.16j has introduced the concept of mesh network model and a special type of node called Relay Station (RS) for traffic relay for Subscriber Stations (SSs). A WiMAX mesh network is illustrated in Figure 1, which is able to provide larger wireless coverage, higher network capacity and Non-Line-Of-Sight (NLOS) communications [57]. This model is especially suitable for some application scenarios, such as broadband Internet access and emergency communications.

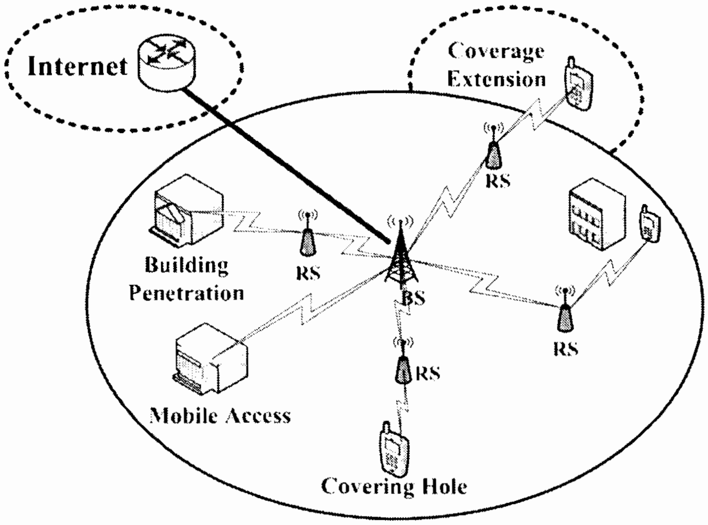


Figure 1 A WiMAX mesh network

This thesis studies a Distance-Aware Relay Placement (DARP) problem in WiMAX mesh networks, which considers a more realistic model that takes into account physical constraints such as channel capacity, signal strength and network topology, which were largely ignored in previous studies. The goal here is to deploy the minimum number of RSs to meet system requirements such as user data rate requests, signal quality and network topology. I divide the DARP problem into two sub-problems, *LOwer-tier Relay Coverage* (LORC) Problem and *Minimum Upper-tier Steiner Tree* (MUST) Problem. For LORC problem, I present two approximation algorithms based on independent set and hitting set, respectively. For MUST problem, an efficient approximation algorithm is provided and proved. Then, an approximation solution for DARP is proposed and proved which combines the solutions of the two sub-problems.

Another important challenge is the multi-hop scheduling scheme for the network. The physical layer of WiMAX uses Orthogonal Frequency-Division Multiple Access (OFDMA) since OFDM has two-fold benefits in terms of robustness to multi-path fading, and ease of digital signal processing implementation. Thus, this thesis also studies a Multi-hop Fair Scheduling for Throughput Optimization (MFASTO) problem in WiMAX mesh networks. The goal here is to maximize the minimum satisfaction ratio among all the SSs. In order to solve the MFASTO problem, an ILP formulation and an efficient heuristic algorithm are proposed in this work. I also present numerical results confirming the theoretical analysis of our schemes as the first solution for the DARP problem and the MFASTO problem.

Through extensive research has been carried out studying the scheduling problem in wireless networks, the scheduling problem in the overwater wireless networks has not been well studied. Wireless communications over water may suffer from serious multipath fading due to strong specular reflections from conducting water surfaces. Cognitive radios enable dynamic spectrum access over a large frequency range, which can be used to mitigate this problem. In this thesis, I studied how to leverage cognitive radios for effective communications in wireless networks over water. I formally define the studied problem as the Overwater Radio-Time Scheduling (OVERTS) problem which seeks a *radio channel with time* schedule such that the total of assigned *eligible time slots*, in which a “good” communication link is maintained between every Mobile Station (MS) and the Base Station (BS), satisfies the time slots requirement of each MS. Two effective heuristic algorithms are presented for the OVERTS problem. Simulation results are presented to justify the performance and efficiency of our proposed scheduling algorithms.

The rest of this thesis is organized as follows. I studied the distance-aware relay placement problem in WiMAX mesh networks in Chapter 2. Then, I studied the max-min fair scheduling problem in OFDMA-based WiMAX mesh networks in Chapter 3, which is followed by the study and discussion of cognitive radio scheduling in overwater wireless networks in Chapter 4. Finally, I conclude this thesis in Chapter 5.

CHAPTER 2. DISTANCE-AWARE RELAY PLACEMENT IN WIMAX MESH NETWORKS

The WiMAX technology which is the fourth generation (4G) standard for low-cost, high-speed and long-range wireless communications uses large chunks of spectrum (10-20 MHz or more), and delivers high bandwidth (up to 75 Mbps). Despite the high bandwidth promised by WiMAX, there are several challenges to be solved. The first challenge is to eliminate or reduce *coverage holes*. Because of high path-loss, and shadowing due to obstacles such as large buildings, trees, tunnels, etc., there would be some spots with poor connectivity, which we call coverage holes. This leads to degradation in overall system throughput. Another key design challenge is *range extension*. At times, it is required to provide wireless connectivity to an isolated area outside the reach of the nearest Base Station (BS). To solve the coverage holes and range extension problems, adding more base stations would be an easy choice. However, given the high cost of deploying BSs, such a solution could be overkill, and too expensive [5]. In such contexts, relay stations (RSs) are a cost-effective alternative. Recently, IEEE 802.16j [59] has been proposed to enhance the existing standard IEEE 802.16e [58], which introduces the concept of mesh network model and a special type of node, *relay station* for traffic relay for Subscriber Stations (SSs). RSs act as MAC-layer repeaters to extend the range of the base station. An RS decodes and forwards MAC-layer segments unlike a traditional repeater which merely amplifies and retransmits PHY-layer signals. Hence, an RS may use a different modulation coding scheme for reception and forwarding of a MAC segment.

This chapter studies the RS placement problem for the models where the locations of SSs are known and the placement of RSs can be controlled to meet data rate, link capacity or signal quality requirements.

Related Works

Relay station placement has been an active research topic in wireless networks, especially in wireless sensor networks. By using RSs, one could deploy a network at a lower cost than using only (more expensive) BSs to provide wide coverage while delivering a required level of service to users [17], [26], [27], [48]. Relay node placement problems are usually classified into two classes: single-tiered (both relay nodes and sensor nodes can relay traffic) and two-tiered (sensor nodes cannot relay traffic). In [28], Lin and Xue proved the single-tiered placement problem with $R = r$ and $K = 1$ is NP-hard, where R, r and K denote the transmission range of relay nodes, the transmission range of sensor nodes, the connectivity requirement respectively. A 5-approximation algorithm was presented to solve the problem. The authors also designed a *steinerization* scheme which has been used by many later works. Better constant factor approximation algorithms for the cases where $R \geq r$ and/or $K > 1$ have been presented in [29], [55]. In [14], a 3.11-approximation algorithm was presented. The authors also proved that one-tier version admits no Polynomial Time Approximation Scheme (PTAS), assuming $P \neq NP$. For the *two-tiered placement*, under the assumption that $R \geq 4r$, a 4.5-approximation algorithm was provided in [45]. Lloyd and Xue [29] relaxed the assumption and presented a $(5+\grave{o})$-approximation algorithm for the problem with $R \geq r$ and $K = 1$. [14] improved the approximation algorithm in [29] by providing a PTAS. A $(10+\grave{o})$-approximation algorithm has been presented in [55] for the case where $R \geq r$ and $K = 2$. In [18], the authors studied a fault-tolerant relay placement problem in heterogeneous sensor networks, where target nodes have different transmission radii. However, the work still assumed that the transmission range of relay nodes is the same.

Besides minimizing the number of placed RSs, some work has been done on placement with physical constraints, such as energy consumption and network lifetime. Hou *et al.* studied the energy provisioning problem for a two-tiered wireless sensor network [21]. Besides provisioning additional energy on the existing nodes, they consider deploying relay nodes (RNs) into the network to mitigate network geometric deficiency and prolong network lifetime. In [49], Hassanein *et al.* proposed three random relay deployment strategies for connectivity-oriented, lifetime-oriented and hybrid deployment. In [35], Pan *et al.* studied base station placement to maximize network lifetime. Along this line, [40] considered joint base station placement and data routing strategy to maximize network lifetime. The same group studied using mobile base stations to prolong sensor network lifetime in [41].

Comprised of small form factor low-cost relays, associated with specific BSs, the main advantages of the WiMAX relay network model are increased coverage and capacity enhancement [30]. RSs are expected to have significantly lower complexity than 802.16e BSs. In [26], an optimal scheme was proposed to find the location of a single RS and resource allocation for all the SSs. In [27], the authors introduced a novel dual-relay architecture, where each SS is connected to the BS via exactly two active RSs through the decode-and-forward scheme. They proposed a two-phase heuristic algorithm to solve the dual-relay RS placement problem. The authors of [52] divided the network into clusters.

Then in each cluster, Integer Linear Programming (ILP) formulation was proposed to select the locations for BSs and RSs from a set of given positions. Recently, new dual-relay coverage architecture was proposed for 802.16j Mobile Multi-hop Relay-based (MMR) networks [26], [27], where each subscriber station (SS) is covered by two RSs. [26] assumed that only one RS is placed in each cell. ILP formulation was applied to find an optimal placement of RS which can maximize the cell capacity in terms of user traffic rates. In [27], assuming a uniform distribution on user traffic demand, the authors studied how to determine the RSs' locations from a set of predefined candidate positions.

Problem Statement

According to IEEE 802.16j [60], a WiMAX mesh network is composed of a BS, SSs and a set of RSs. An RS can relay traffic for SSs and the BS, and an SS does not have the routing and traffic relay capabilities. This communication scenario is worth studying since in the near future, there may exist a large number of simple WiMAX terminals (SSs) in need of network connections, just like the current WiFi terminals. As suggested by the WiMAX standard [58], [59], a tree rooted at the BS is usually constructed to support packet forwarding in a WiMAX mesh network. In the tree, all SSs must be leaf nodes and only the RSs can serve as intermediate (non-leaf) nodes connecting the SSs with the BS. By placing the RSs in the network, we actually construct a tree structure and a routing strategy for the WiMAX network. It has been shown that RS placement has a significant impact on network performance [26], [32].

A. Network and Relay Models

In this thesis, IEEE 802.16j Mobile Multi-hop Relay (MMR) network is used as the model for the network infrastructure. As proposed in the IEEE 802.16j standard [60], an 802.16j radio link between a BS and an RS or between a pair of RSs is a *relay link*. Concatenation of k consecutive relay links ($k \geq 1$) between the BS and the designated access RS forms a relay path. Compared to the BS, an RS has a significantly simpler hardware and software architecture, and hence a lower cost. An RS merely acts as a link layer repeater, and therefore does not require a wired backhaul. Furthermore, an RS needs not perform complex operations such as connection management, hand-offs, scheduling, etc. Also, an RS typically operates at a much lower transmit power, and requires lower-MAC and PHY-layer stack. All these factors lead to a much lower cost of an RS, and thus, relay networks are evolving as a low-cost option to fill coverage holes and extend range in many scenarios.

The major goals of deploying relay stations are to improve coverage in geographic areas that are severely shadowed from the BS, to extend the range of a BS, and to improve the link data rate and network throughput. IEEE 802.16j [60] defines three types of RSs whose functions are to relay traffic between an SS and a BS, including Fixed Relay Station (FRS), Nomadic Relay Station (NRS), and Mobile Relay Station (MRS). An FRS is a relay station that is permanently installed at a fixed location. An NRS is a relay station that is intended to function at a fixed location for periods of time comparable to a user session. An MRS is a relay station that is intended to function while in motion. In this work, we consider static SSs such as McDonald's, gas stations, and grocery stores. Thus, I will study a *static network planning* problem, i.e., finding where to place a minimum number of relay nodes such that certain performance requirements can be satisfied. Therefore, I focus on FRSs and NRSs in this work. In this chapter, I study the *two-tiered* relay station placement problem which is particularly suitable for WiMAX-based mesh networks. The two-tiered network model divides the network into two tiers, as shown in Figure 2.

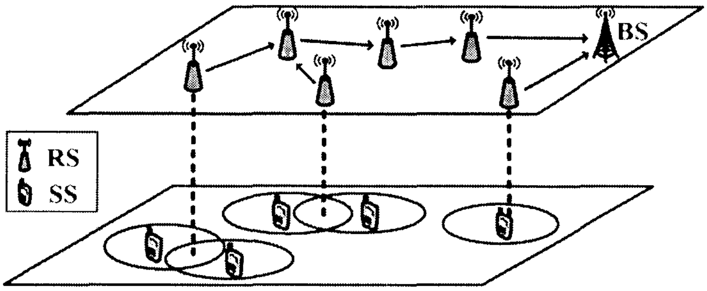


Figure 2 Two-tiered relay model

All the SSs form the lower tier, each of them is covered by at least one RS, through which each SS can relay its traffic cooperatively to the BS. Meanwhile, following the WiMAX mesh network convention, all the RSs and the BS are connected on the upper tier to enable two-hop or multi-hop relay capability.

B. Distance-Aware Relay Station Placement

Because user data rate requests, channel capacity and the LOS effect should be carefully taken into account for the RS placement, I will study an RS placement problem satisfying each SS's data rate request, which has not been well addressed before. Note that previous studies on relay node placement have mainly focused on coverage and connectivity.

Definition 1 (Feasible coverage). Let s_i be a fixed SS with known location, and b_i be its data rate request (in terms of *bps*). An RS r_m is said to provide *a feasible coverage* for s_i if the channel capacity of the link (in terms of *bps*) between s_i and r_m is sufficient for the data rate request of s_i. In other words, the capacity of link (s_i, r_m) is no less than b_i.

Two kinds of placement scenarios are defined in WiMAX standards: *two-hop relay* and *multi-hop relay*. According to the IEEE 802.16j [60], supporting 2-hop relay is mandatory but supporting multi-hop relay (more than 2) is optional. In this chapter, I study the *multi-hop relay* for the RS placement, while most previous work studies the *two-hop relay* model.

It is well-known that the capacity of a wireless connection is highly related to the Euclidean distance between its two end nodes [11]. If the two-ray ground path loss model is considered (which is generally used for modeling the large scale signal strength over the distance of several kilometers that use tall towers as well as for LOS micro-cell channels) [26], the power level at the receiver P_r is given as

$$P_r = P_t G_t G_r h_t^2 h_r^2 d^{-\alpha} \tag{1}$$

where P_t is the transmission power, and G_t / G_r and h_t / h_r are the gains and heights of transmitter antenna and receiver antenna, respectively. d is the Euclidean distance between the transmitter and the receiver, and α is the attenuation factor, which depends on the environment and typically varies in a range of 2 − 4 for the terrestrial propagation. Then the signal-to-noise ratio (*SNR*) at receiver is $SNR_r = P_r / N_0$, where N_0 is the thermal noise power at the receiver which is normally a *constant*. Based on Shannon's theorem, the link capacity is given by $W \log(1 + SNR_r)$, where W is spectrum bandwidth. Therefore, when the noise N_0 is constant, the received signal quality, and consequently *the channel* *capacity, are determined by the received signal strength* P_r. The problem I study is a special case where the channel capacity of each link between each SS and its corresponding RS is *decided by the received power at the receiver* P_r.

Based on formula (1), for each RS or SS, its transmitter/receiver gain is set to be fixed. We can see that for a pair of transmitter and receiver, the signal received at the receiver is decided by *the distance between the pair*. Consequently, the channel capacity of the transmission between an SS and its covering RS is decided by the distance between these two stations. Therefore, *the data rate request* b_i *of each SS* s_i *can be translated into an equivalent problem with requirement of distance between* s_i *and its covering RS.*

Definition 2 (Distance-Aware Relay Placement (DARP) Problem). Given a WiMAX mesh network with a BS and a set of SSs $S = \{s_1, s_2, \ldots, s_n\}$, let $D = \{d_1, d_2, \ldots, d_n\}$ be the distance requirement set for the SSs. The DARP problem seeks a *minimum number* of RSs R such that:

(1) Providing feasible coverage for each $s_i \in S$. In other words, s_i is covered by at least one RS or the BS within distance d_i

(2) Each placed RS has enough data rate to relay traffic for each SS or another RS that it covers for relay.

In addition to the feasible coverage of the SSs in the lower-tier (Condition 1), to ensure that all the packets from SSs can transmit to the BS, we need to consider the connectivity of the placed RSs and the BS through possible multi-hop relays in the upper-tier (Condition 2).

Approximation Solution for DARP Problem

When all the distance requirements are the same ($d_1 = d_2 = \ldots = d_n$), DARP becomes the 2tRNP problem in [29], which was proved to be *NP*-hard [16]. Thus, DARP is *NP*-hard. Given the hardness of the problem, it is not possible to find a polynomial time optimal solution for DARP unless $P = NP$ [16]. Therefore, the best solutions we can expect are polynomial time approximation algorithms. To solve the DARP problem, I divide the problem into two sub-problems and conquer them one by one. First, I focus on the lower tier and aim to find a minimum set of RSs for feasible coverage of the SSs. Next, I move onto the upper tier and provide distance-constrained connections between RSs and the BS. We discuss each sub-problem in the following.

A. Lower-tier Coverage of Subscriber Stations

In the first step, we need to solve the coverage sub-problem in the lower tier, which seeks to use the minimum number of RSs to guarantee that each SS is covered feasibly.

Definition 3 (LOwer-tier Relay Coverage (LORC) Problem). Given a WiMAX mesh network with a BS and a set of subscriber stations $S = \{s_1, s_2, \ldots, s_n\}$, let $D = \{d_1, d_2, \ldots, d_n\}$ be the distance requirement set for the SSs. The LORC problem seeks a *minimum number* of relay stations R that provides feasible coverage for each $s_i \in S$. In other words, s_i is covered by at least one RS or the BS within distance d_i .

When all the SSs have the same distance requirement, it is easy to see that the coverage problem is equivalent to the GeoDC problem [15], which seeks a minimally sized set of disks (of prescribed radius) covering all points in a Euclidean plane [20]. Therefore, with a special case being *NP*-hard, we can see that LORC is also *NP*-hard.

1) Maximal Independent Set Based Approximation Solution:

Our first solution is based on the following observations. First, to provide a feasible coverage for an SS s_i with distance requirement d_i, it is easy to see that an RS must be placed *in* or *on* a disk centered at s_i with radius d_i. I denote such a disk by the *feasible coverage disk* for s_i. Second, for any two SSs, if their feasible coverage disks intersect with each other, then they can be covered by one RS in the intersection area. We called such two SSs are *neighbors*. Similarly, if multiple SSs are all neighbors with each other (a *clique*), all these SSs can be covered by one RS. Based on above observations, I present a *simple and provably good solution in* Algorithm 1.

Algorithm 1 LORC-MIS (S, D)

```
Construct sets of disks $C$ and $C'$; $C \leftarrow \emptyset; C' \leftarrow \emptyset$
for $s_i \in S$ do
    Calculate its feasible coverage disk $C_i$; $C \leftarrow C \cup \{C_i\}$.
end for
while $C \neq \emptyset$ do
    Find $s_{min}$, the SS with the minimum distance requirement $d_{min}$; $C_{min}$ is the feasible coverage disk of $s_{min}$; $C' = \{C_{min}\}$;
    Construct a regular hexagon $H_{min}$ centered at $s_{min}$ with side length $\sqrt{3}d_{min}$;
    Construct a point set $P = \{6$ vertices of $H_{min}$, $s_{min}\}$;
    for $C_i \in C$ do
        if $C_i$ interests with $C_{min}$ then
```

```
11:             $C' \leftarrow C' \cup \{C_i\}$
12:         end if
13:     end for
14:     while $C' \neq \varnothing$ do
15:         Choose the point $v \in P$ which covers most disks in $C'$;
16:         Place an RS $r$ at the location of $v$; $R \leftarrow R \cup \{r\}$;
17:         Remove all the disks covered by $r$ from $C$ and $C'$;
18:     end while
19: end while
20: return $R$
```

Let us use an example from Figure 3 to Figure 6 to illustrate our algorithm. For SSs s_1, s_2, s_3, s_4, and s_5 in Figure 3, we first calculate their respective feasible coverage disks C_1, C_2, C_3, C_4, and C_5 (Lines 2 - 4), shown by the circles around each node.

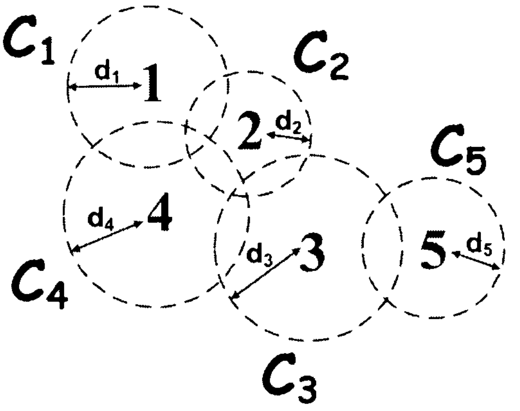


Figure 3 Users and coverage disks

Figure 4 demonstrates the neighboring relationship between two nodes if two disks intersect each other. We first select disk C_2 which has the smallest radius (Line 6).

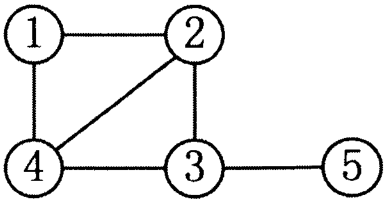


Figure 4 Neighboring graph

Next, as shown in Figure 5, we construct a regular hexagon H_2 for C_2, and then have set P including 7 possible positions $\{H_2^A, H_2^B, H_2^C, H_2^D, H_2^E, H_2^F, s_2\}$ to place RSs (Lines 7 - 8). From Line 9 to Line 13, we calculate $C' = \{C_1, C_2, C_3, C_4\}$, which are the disks that will be covered in this step.

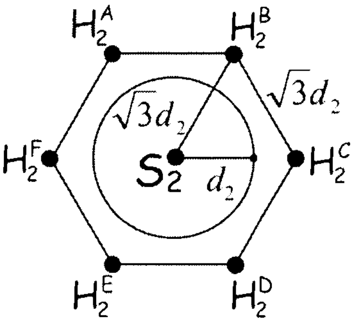


Figure 5 Regular hexagon

In Figure 6, we first select H_2^F to place an RS because it can cover most (two) disks (SSs), C_1 and C_4 (Line 14 to Line 18). Then, two disks $\{C_2, C_3\}$ are left in C' to be covered. Following same process, an RS is place on s_2, and another is placed on H_2^D to cover these two disks. At this time, after removing SSs 1, 4, 2, and 3, $C = \{C_5\}$, which is not empty. Similarly, we construct a regular hexagon H_5 for C_5, place an RS at the center of H_5, and remove SS 5. The solution uses four RSs $R = \{s_2, s_5, H_2^F, H_2^D\}$ to cover all SSs.

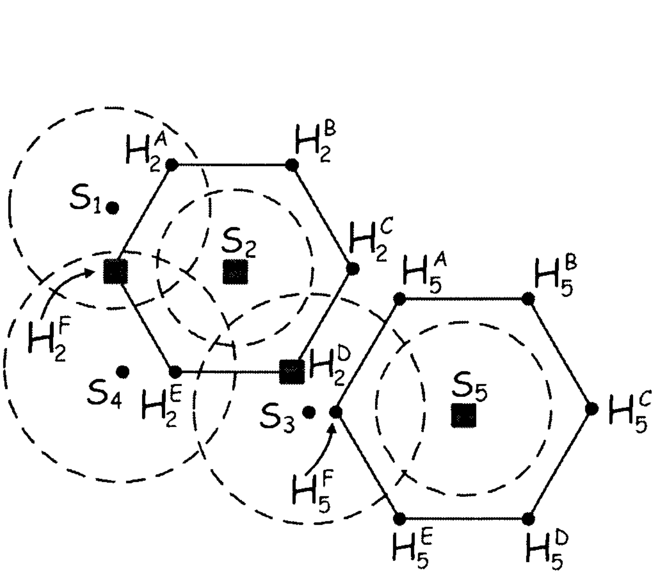


Figure 6 Relay placement

Next, we aim to prove that our RS placement algorithm is actually a 7-approximation solution for the LORC problem.

Lemma 1. Given an SS s_{min}, which has the smallest radius d_{min}, and a set $N_{s_{min}}$ including all the neighboring SSs of s_{min}, $N_{s_{min}} \cup \{s_{min}\}$ can be covered by *at most* 7 RSs.

Proof: For any neighbor SS o of s_{min}, assuming its distance requirement is $d_o(\geq d_{min})$, without loss of generality, o is located in the region of angle $\angle H^A_{s_{min}} s_{min} H^B_{s_{min}}$, where $H^A_{s_{min}}$ and $H^B_{s_{min}}$ are two adjacent vertices of hexagon $H_{s_{min}}$. Now we want to prove that there must be a point $v \in \{H^A_{s_{min}}, H^B_{s_{min}}, s_{min}\}$ that can cover SS o. In other words, v is in or on disk C_o, the feasible coverage disk of o.

Let us use the auxiliary graphs in Figure 7 and Figure 8 to illustrate our proof. For the simplicity, I use a, b, and s to denote $H^a_{s_{min}}$, $H^B_{s_{min}}$, and s_{min}, respectively. I first construct perpendiculars at node a on line $s-a$, and at node b on line $s-b$, respectively.

The two perpendiculars intersect at point q. Based on geometrical properties, we can derive that $\angle asq = \angle bsq = 30°$ and $\angle aqs = \angle bqs = 60°$.

Without loss of generality, I assume that o lies in the region of $\angle asq$. Now we need to prove that o can be covered by an RS on a or s. In Figure 7 and Figure 8, I draw a circle C_a centered at a with radius d_{min}. There are two cases to consider.

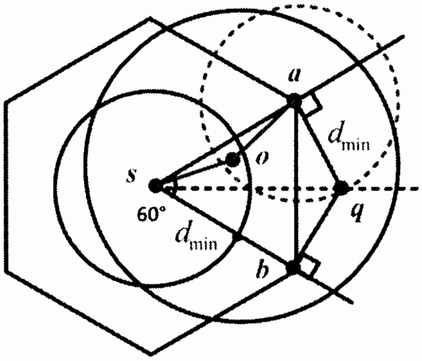


Figure 7 *o* in/on disk *Cs* or *Ca*

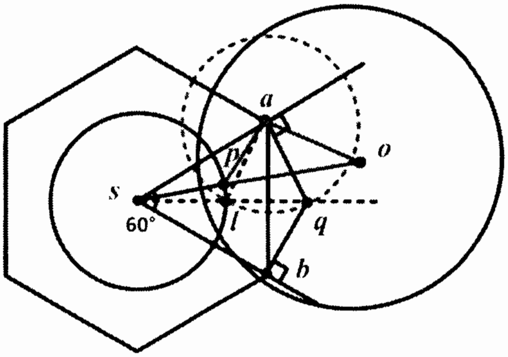


Figure 8 *o* not in/on disks *Cs* and *Ca*

CASE 1 - o is in or on disk C_s or disk C_a: As shown in Figure 7, if o is in or on disk C_a, we can easily see that $|so| \le d_{min} \le d_o$. Therefore, s is in or on the disk of C_o and can cover SS o. Similarly, if o is in (on) the disk C_a, then $|ao| \le d_{min} \le d_o$. Thus, a is in (on) the disk of C_o and covers o.

CASE 2 - o is NOT in or on disk C_s and disk C_a: As shown in Figure 8, if o is in or on neither C_s nor C_a, we need to prove that it can be covered by a. To prove it, we need two auxiliary lines; one connects s and o. This line intersects with disk C_s at point p. The other connects s and q, which intersects with C_s at point l. Since $\angle aql = \angle aqs = 60°$, and

$$|ql| = |sq| - |sl| = 2d_{min} - d_{min} = d_{min} = |qa| \qquad (2)$$

it is easy to see that triangle $\square alq$ is an equilateral triangle. Thus, $\angle alq = \angle laq = 60°$. It is easy to see that

$$\angle apo \le \angle alq = \angle laq \le \angle pao \qquad (3)$$

Therefore, in $\square apo$, we have $|ao| \le |po|$. Meanwhile, we have

$$|sp| + |po| = |so| \le d_{min} + d_o \qquad (4)$$

Given $|sp| = d_{min}$, we know that $|po| \le d_o$. And consequently, $|ao| \le d_o$, and a could be used to cover o.

Combing both cases, any neighbor o of s can be covered by either s or one of the six vertices of hexagon H_s. Therefore, at most 7 RSs can cover s and all of its neighbors.

Theorem 1. Algorithm 1 is a 7-approximation for the LORC problem. More specifically, if the number of the RSs returned by Algorithm 1 is denoted by $|R|$, we have $|R| \le |OPT_L|$, where OPT_L is an optimal solution for LORC.

Proof: At Line 6 in Algorithm 1, each time we select a remaining uncovered SS with the minimum distance requirement, and try to cover it as well as all of its neighboring

SSs. From Lemma 1, we know that, at each time, at most 7 RSs are needed to cover a selected SS and all its neighbors. Assume that the total number of SSs selected in this step is L, then the total number of RSs will be no more than $7L$.

Meanwhile, using the neighboring graph in Figure 4, we can see that these L nodes form a maximal *independent set* of the graph. Denote a maximum *independent set* by M with size W, it is obvious that *no any two or more nodes in* M *can be covered by one RS.* In other words, each node in M needs one RS to exclusively cover itself. Thus, W RSs have to be placed to cover the nodes in M. Since M is a subset of G, in order to cover all the nodes in G, it is easy to see that at least W RSs are needed. Therefore, we have $|OPT_L| \geq W$. Consequently, the number of RS placed by Algorithm 1 is

$$|R| \leq 7L \leq 7W \leq 7|OPT_L| \tag{5}$$

Therefore, Algorithm 1 is a 7-approximation.

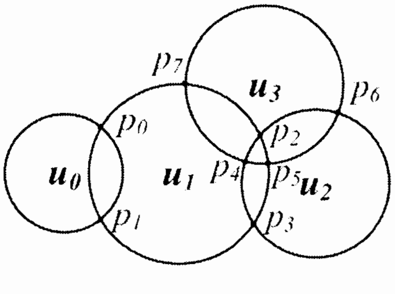
2) Hitting Set Based Approximation Solution:

In this section, I want to improve our solution by exploring the geometric structure of the problem. Our solution is based on the relationship between LORC and the well-known *hitting set* problem.

Definition 4 (Hitting set problem). Given a set $S = \{e_0, e_1, \ldots, e_m\}$ and a collection of sets $C = \{S_i \mid 0 \leq i \leq n\}$, where S_i is a set of elements $S_i = \{e_j \mid 0 \leq j \leq m\}$, a sub-set $S' \in S$ which contains at least one element from each subset S_i in C is a *hitting set.* A hitting set with the smallest size is the *minimum hitting set.*

For example, given $S = \{0,1,2,3,4\}$ and a collection of sets $C = \{\{0,1\},\{1,2,3\},\{3,4\}\}$, a minimum hitting set is $\{1,3\}$. Finding a minimum hitting set is a *NP*-hard problem [16]. There exist efficient approximation algorithms [10], [36] for the hitting set problem, and a PTAS for geometric hitting set problem [31].

To see how to translate LORC into an equivalent hitting set problem, I use an example in Figure 9 for illustration. 4 users u_0, u_1, u_2 and u_3 are to be covered. u_0 and u_1 are neighbors, C_{u_0} and C_{u_1} intersect on points p_0 and p_1. u_1 and u_2 are neighbors, C_{u_1} and C_{u_2} intersect on points p_2 and p_3. C_{u_3} intersects with C_{u_1} and C_{u_2} at p_5, p_7 and p_4, p_6, respectively. It is easy to see that by placing an RS on $\{p_0, p_1\}$, user u_0 will be covered. Similarly, an RS on any location from $\{p_2, p_3, p_4, p_5, p_6\}$ will cover u_2. Thus, I construct an instance of hitting set from an instance of LORC by giving set $S = \{p_0, p_1, p_2, p_3, p_4, p_5, p_6, p_7\}$ and a collection of sets $C = \{S_0, S_1, S_2, S_3\}$, while $S_0 = \{p_0, p_1\}$, $S_1 = \{p_0, p_1, p_2, p_3, p_4, p_5, p_7\}$, $S_2 = \{p_2, p_3, p_4, p_5, p_6\}$, and $S_3 = \{p_2, p_4, p_5, p_6, p_7\}$, respectively. If we find a minimum hitting set $\{p_0, p_2\}$, then by placing RSs at p_0 and p_2, we will cover all the users.



$S = \{p_0, p_1, p_2, p_3, p_4, p_5, p_6, p_7\}$
$C = \{S_0, S_1, S_2, S_3\}$

$S_0 = \{p_0, p_1\}$
$S_1 = \{p_0, p_1, p_2, p_3, p_4, p_5, p_7\}$
$S_2 = \{p_2, p_3, p_4, p_5, p_6\}$
$S_3 = \{p_2, p_4, p_5, p_6, p_7\}$

Figure 9 Relationship between LORC and minimum hitting set

Based on our observation, I present a simple Hitting Set Based algorithm to solve the LORC Problem. This algorithm is formally presented by Algorithm 2.

```
Algorithm 2 LORC-HS ($S$, $D$)
1: for any two SSs $s_i$ and $s_j$ do
2:     Calculate its feasible coverage disks $C_i$ and $C_j$;
3:     if $C_i$ and $C_j$ intersect with each other then
4:         Assume the intersection points are $p$ and $q$ (or $p = q$);
5:         $I = I \cup \{p, q\}$;
6:     end if
7: end for
8: for each SS $s_i$ do
9:     $H_i = \varnothing$;
10:    for each point $p \in I$ is in or on disk $C_{s_i}$ do
11:        $H_i = H_i \cup \{p\}$;
12:    end for
13: end for
14: Construct a set $H = \{H_0, H_1, \ldots, H_n\}$;
15: $H_{min} \leftarrow$ Solve a minimum hitting set problem MHS($H$);
16: Place an RS on each point $p \in H_{min}$
```

In Algorithm 2, we first find the set (I) of all possible RS locations, which are the intersection points of covering disks (Line 1 - Line 7). Then, we construct an instance of hitting set problem from the instance of LORC. For each s_i, we try to find a corresponding set H_i in the instance of hitting set, which includes all the positions that can cover s_i (from Line 8 to Line 13). Note that the geometric hitting set problem admits a PTAS [31]. It is easy to see that Algorithm 2 could return a $(1 + ò)$-approximation scheme for LORC.

B. Upper-Tier Connectivity of Relay Stations

Besides the coverage of the SSs in the lower tier, another important requirement for RS placement is the connectivity between the RSs and the BS, which promises the connections from SSs to the BS. After the coverage stage, in the upper-tier, if the BS and the covering RSs are all connected, then we already have a solution. However, if they are not connected, which is more typical given the large range of a WiMAX cell, we need to study the problem of how to connect the BS and the RSs. The basic idea of providing connectivity is to add more RSs for *multi-hop* relay. In the upper tier, I aim to construct a tree-topology, where BS is the root, all the coverage RS placed for the lower tier SSs are the leaf nodes, and the newly added RSs will be the intermediate nodes on the tree. If we regard the coverage RSs and the BS as target points, then the upper-tier connection problem is related to the well-known constrained Steiner tree problem [16], [28].

Definition 5 (Minimum Upper-tier Steiner Tree (MUST) problem). Given $X = \{x_1, x_2, \ldots, x_n\}$ be the set of n target points (which are the BS and coverage RSs placed for SSs in LORC), MUST seeks a constrained steiner tree T spanning the set X of target points and a set of *minimum* additional steiner points (new RSs to be placed) such that:

- *Each tree edge length should be no more than D_i, which is the feasible distance requirement for each RS r_i*

It is easy to see that MUST is *NP*-hard given that its special case in [28] is *NP*-hard. In this chapter, I will design an efficient approximation algorithm for the MUST problem based on a concept known as *steinerization*, which was first introduced in [28]. The biggest challenge is that the newly placed RSs will have *various* distance requirements. Most previous works assume that all the SSs have the same transmission range, and all the RSs have the same transmission range [28], [29], [55]. In [18], the authors studied to provide the single-tiered 2-connectivity placement for all the terminals that have different transmission ranges. However, they still assumed that RSs share the same range.

To solve MUST, the first challenge is *how to decide the distance requirement for each RS*, which is affected by the SSs or RSs being covered. In order to guarantee the data rate of each SS, for each RS r_i, the link capacity between r_i and its parent node on the tree T cannot be lower than the one between r_i and any child of its. The definition of distance requirement of RS is formally given in the following.

Definition 6 (Distance Requirement of RS). For each RS r_i, Di, the distance requirement of r_i, which represents the maximum feasible distance between r_i and its parent on tree T, equals to the minimum distance requirement of all its children. In other words, $D_i = \min_{k \in \mathrm{T}_i} d_k$, where T_i is the sub-tree of T rooted at r_i.

An example is shown in Figure 10 for demonstration. RS_1 covers two SSs A and B, whose distance requirements are 16 and 15, respectively. Therefore, the distance requirement of RS_1 is 15, which can guarantee that the data rate requirements of A and B can be satisfied. Similarly, RS_2 has its own distance requirement of 18. For RS_3, its distance requirement is 15, which is the smallest among D_{RS_1} and D_{RS_2}. It is worth noting that this work studies *ensuring data rate for each individual* SS or RS. With the approach to deciding distance requirements, our solution for MUST is listed in Algorithm.3.

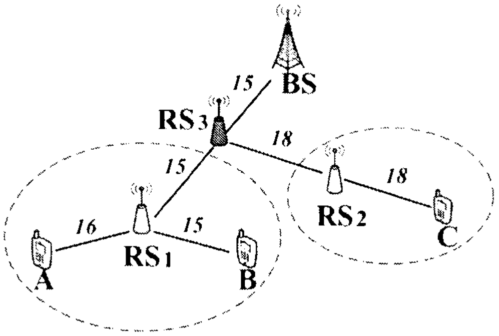


Figure 10 Example of distance requirement of RSs

Algorithm 3 MUST (X, D)

```
Construct a complete graph $G = (X, E)$; $d_{min} = \min_{i \in S} d_i$;
for each edge $e(x_i, x_j)$ do
    Assign weight $w_1(x_i, x_j) = \left\lceil \frac{\| e(x_i, x_j) \|}{d_{min}} \right\rceil - 1$ on the edge;
end for
Find a minimum spanning tree $T_{mst}$ of $G$ with BS as the root;
for each RS $x_i$ do
    Calculate the distance requirement $D_i = \min_{x_j \in T_i} d_j$;
end for
for each RS $x_i$ and its parent $x_i^p$ on $T_{mst}$ do
    $w_2(x_i^p, x_i) = \left\lceil \frac{\| e(x_i^p, x_i) \|}{D_i} \right\rceil - 1$;
    Place $w_2(x_i^p, x_i)$ RSs on edge $e(x_i^p, x_i)$ separating the edge into $\left\lceil \frac{\| e(x_i^p, x_i) \|}{D_i} \right\rceil$ parts
    with each node having feasible distance;
end for
```

Let us use an example from Figure 11 to Figure 13 to illustrate our solution. The network includes BS, RS_1 and RS_2. First, we construct an undirected complete graph in

Figure 11 (Line 1). We then assign edge weight $w_1(e) = \lceil \| e \| / d_{min} \rceil - 1$ on each edge e (Lines 2 - 4), where d_{min} is the minimum distance requirement among all the nodes, which is 5 in the example. The distances of edges (BS, RS_1), (BS, RS_2) and (RS_1, RS_2) are 20, 21 and 16, respectively. The corresponding weights of these edges are 3, 4 and 3. Next in Line 5, a minimum spanning tree is constructed in Figure 12. Now we have the parent-child relationship between nodes. For example, RS_1 is the parent of RS_2. Based on the parent-child relation, D_i for each RS r_i has to be updated (Lines 6 - 8). For example, RS_1 has to reduce its distance requirement to 5 to ensure the service for its child RS_2.

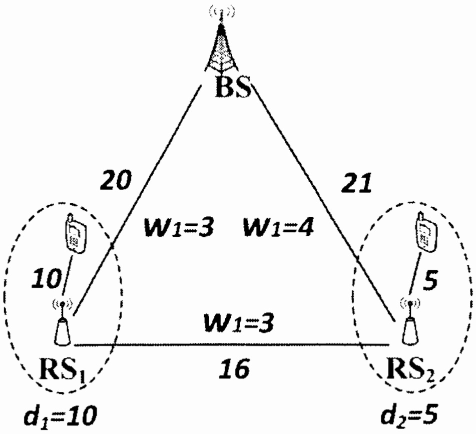


Figure 11 Complete graph

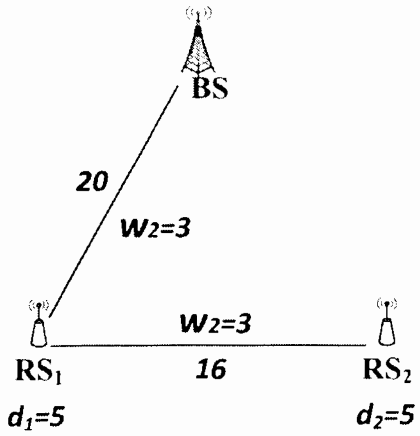


Figure 12 Minimum spanning tree

Next, we need to re-calculate the edge weight $w_2(e)$ for each tree edge e (Line 10), shown in Figure 12, and then place RSs accordingly (Line 11), shown in Figure 13.

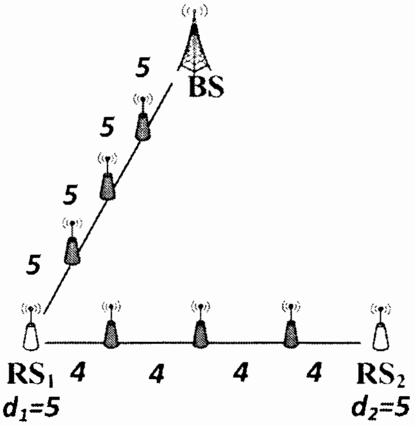


Figure 13 MUST solution

Theorem 2. Algorithm 3 finds an $\frac{8 \cdot d_{max}}{d_{min}}$-approximation for the MUST problem. In other words, let R_M be the set of RSs placed by our solution and OPT_M be an optimal solution for MUST, we have

$$|R_M| \leq \frac{8 \cdot d_{max}}{d_{min}} |OPT_M| \qquad (6)$$

where d_{min} and d_{max} denote the *minimum* and *maximum* distance requirements from SSs, respectively.

Proof: I assume that each edge needs at least one RS placed (if no RS is needed on an edge, I ignore this edge because it does not affect the solution). First, I consider a *special case*, MUST(X , d_{min}) (denoted by ST) that all users have the same distance requirement d_{min}. ST is a *Minimum Steiner Tree* problem studied in [9], [28]. I denote R_{ST} , OPT_{ST} as our scheme and an optimal solution for the ST problem, respectively.

Given OPT_M, an optimal solution for MUST(X, D), instead of placing RSs with distance D_i, we place RSs with distance d_{min} on the same tree structure. Then we will have a *feasible solution*, denoted by OPT'_M, for ST. The number of RSs placed on each edge e changes from $\left\lceil \frac{\|e\|}{D_i} \right\rceil - 1$ to $\left\lceil \frac{\|e\|}{d_{min}} \right\rceil - 1$. Therefore,

$$|OPT_{ST}| \le |OPT'_M| = \frac{\left\lceil \frac{\|e\|}{d_{min}} \right\rceil - 1}{\left\lceil \frac{\|e\|}{D_i} \right\rceil - 1} |OPT_M| \qquad (7)$$

Let $\|e\| = \alpha_i D_i + \beta_i = \alpha_{min} d_{min} + \beta_{min}$, where $\alpha_i, \alpha_{min} \ge 1, \beta_i < D_i$ and $\beta_{min} < d_{min}$. We have

$$\frac{\alpha_{min}}{\alpha_i} = \frac{D_i}{d_{min}} + \frac{\beta_i - \beta_{min}}{\alpha_i d_{min}} \qquad (8)$$

CASE 1: If $\beta_i > 0$

$$\frac{|OPT_{ST}|}{|OPT_M|} = \frac{\left\lceil \frac{\|e\|}{d_{min}} \right\rceil - 1}{\left\lceil \frac{\|e\|}{D_i} \right\rceil - 1} = \frac{\left\lceil \alpha_{min} + \frac{\beta_{min}}{d_{min}} \right\rceil - 1}{\left\lceil \alpha_i + \frac{\beta_i}{D_i} \right\rceil - 1} \le \frac{\alpha_{min}}{\alpha_i} \qquad (9)$$

If $\beta_i \le \beta_{min}$, then based on Equation (9)

$$\frac{\alpha_{min}}{\alpha_i} \le \frac{D_i}{d_{min}} \le \frac{d_{max}}{d_{min}} \qquad (10)$$

If $\beta_i > \beta_{min}$, then based on Equation (9)

$$\frac{\alpha_{min}}{\alpha_i} \leq \frac{D_i}{d_{min}} + \frac{\beta_i}{\alpha_i d_{min}} \tag{11}$$

Because $\alpha_i \geq 1$ and $\beta_i < D_i$, therefore

$$\frac{\alpha_{min}}{\alpha_i} \leq \frac{D_i}{d_{min}} + \frac{D_i}{d_{min}} \leq 2\frac{d_{max}}{d_{min}} \tag{12}$$

CASE 2: If $\beta_i = 0$

$$\frac{|OPT_{ST}|}{|OPT_M|} \leq \frac{\left\lceil \frac{\|e\|}{d_{min}} \right\rceil - 1}{\left\lceil \frac{\|e\|}{D_i} \right\rceil - 1} = \frac{\left\lceil \alpha_{min} + \frac{\beta_{min}}{d_{min}} \right\rceil - 1}{\left\lceil \alpha_i + \frac{\beta_i}{D_i} \right\rceil - 1} \leq \frac{\alpha_{min}}{\alpha_i - 1} \tag{13}$$

Note that 1 only consider *the edges that have at least one RS placed*, then $\alpha_i - 1 \geq 1$.

Consequently, $\frac{1}{\alpha_i - 1} \leq \frac{2}{\alpha_i}$. So we have

$$\frac{|OPT_{ST}|}{|OPT_M|} \leq 2\frac{\alpha_{min}}{\alpha_i} \tag{14}$$

Based on Equation (9), we have

$$\frac{|OPT_{ST}|}{|OPT_M|} \leq 2\left(\frac{D_i}{d_{min}} + \frac{\beta_i - \beta_{min}}{\alpha_i d_{min}}\right) \tag{15}$$

Because $\beta_i = 0$ and $\alpha_i \geq 2$, therefore

$$\frac{|OPT_{ST}|}{|OPT_M|} \leq 2\frac{D_i}{d_{min}} \leq 2\frac{d_{max}}{d_{min}} \tag{16}$$

Combining **CASE 1** and **CASE 2**, we have

$$|OPT_{ST}| \le \frac{2d_{max}}{d_{min}} |OPT_M| \tag{17}$$

For *ST* problem, we have $|R_{ST}| \le 4|OPT_{ST}|$. Combining with formula (17), we know that

$$|R_{ST}| \le 4|OPT_{ST}| \le \frac{8d_{max}}{d_{min}} |OPT_M| \tag{18}$$

It is easy to see that R_{ST} is a feasible solution for MUST(S, X, D). Note that R_M and R_{ST} use the same minimum spanning tree topology T_{mst}, with different distance requirement set D_i and d_{min}. Therefore,

$$|R_M| \le |R_{ST}| \le \frac{8d_{max}}{d_{min}} |OPT_M| \tag{19}$$

This completes our proof.

C. Approximation Algorithm for DARP Problem

With the approximation solutions for LORC and MUST, we can present an approximation algorithm for the DARP problem, which is listed in Algorithm 4.

Algorithm 4 DARP (S, D)

```
$X \leftarrow$ LORC($S$, $D$); //LORC-MS or LORC can be applied
for each edge $x_i \in X$ do
    Among all the SSs covered by $x_i$, pick $s_i$ with the smallest distance requirement;
    Place an RS $z_i$ on the location of $s_i$; $Z = Z \cup \{z_i\}$;
```

```
end for
$Y \leftarrow$ MUST($Z$, $D$);
return $X \cup Y \cup Z$
```

Theorem 3. The set of relay stations produced by Algorithm 4 is a $(2\alpha+\beta)$-approximation for the DARP problem, where α is the approximation ratio of solution for LORC, and β is the approximation ratio of solution for MUST.

Proof: Denote OPT as the optimal solution for the DARP(S, D) problem, O_L and O_M as the optimal solution (the minimal number of RSs) for the LORC(S, D) and the MUST(Z, D) sub-problems, respectively. It is easy to see that OPT is also a feasible solution for the LORC(S, D) problem, then we have

$$|O_L| \le |OPT| \tag{20}$$

If we provide an α-approximation solution S_{cover} for LORC, then we have

$$|S_{cover}| \le \alpha |O_L| \le \alpha |OPT| \tag{21}$$

Note that OPT is an optimal solution for DARP(S, D), it would be a feasible solution to MUST(S, D). Since $Z \in S$, OPT must be a feasible solution for MUST(Z, D). Therefore, we have

$$|O_M| \le |OPT| \tag{22}$$

If we provide a β-approximation solution S_{con} for MUST, it is easy to see that

$$|S_{con}| \le \beta |O_M| \le \beta |OPT| \tag{23}$$

Since our solution is $S_{cover} \cup S_{con} \cup Z$, and $|Z|=|X|=|S_{cover}|$, the number of placed RSs is

$$|S_{cover} \cup S_{con} \cup Z| \leq |S_{cover}| + |S_{con}| + |Z| = 2|S_{cover}| + |S_{con}| \qquad (24)$$

Hence, we have

$$|S_{cover} \cup S_{con} \cup Z| \leq 2|S_{cover}| + |S_{con}| \leq 2\alpha|OPT| + \beta|OPT| = (2\alpha + \beta)|OPT| \qquad (25)$$

This completes the proof of the theorem.

Note that I provide a general framework for the DARP problem. Within the framework, for given requirements on running times and performances (e.g., *k*-approximation), we can provide various approximation algorithms to solve the two sub-problems and consequently provide different approximation solutions to the DARP problem. For example, using LORCMIS and MUST solutions, we provide a fast $(14 + 8\frac{d_{max}}{d_{min}})$ -approximation for the DARP problem. While using LORC-HS and MUST, a solution with better approximation ratio can be found in a much longer time.

Numerical Results

In this section, I present numerical results to confirm the effectiveness of our solutions. I implemented both the *7-approximation* solution and *hitting set* based scheme for LORC, which are denoted as MIS and HS in the figures, respectively. The solution for MUST was also implemented. All our simulation runs were performed on a 2.8 GHz Linux PC with 1G bytes of memory. As in [45], [55], SSs were uniformly distributed in a square playing ground. One base station was deployed at the center of the field. All the figures illustrate the average of 10 test runs for various scenarios.

First, I illustrate the tree topologies generated by our solutions in Figure 14, Figure 15 and Figure 16. With distance requirements randomly distributed in [100,150], 50 SSs were deployed in a 2000 × 2000 sq. unites. For LORC, I presented MIS, HS, and an optimal solution using Integer Linear Programming (ILP) solved by Gurobi Optimizer [61]. From these figures, I observed that ILP placed the smallest number of RSs. HS not only deployed similar number of RSs, but also generated similar tree topology with the one generated by ILP.

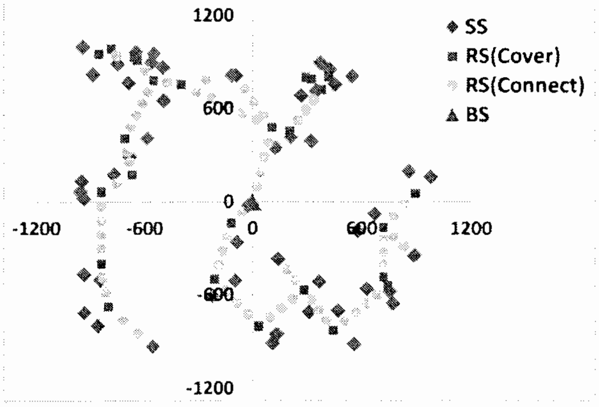


Figure 14 Tree topology (MIS + MUST)

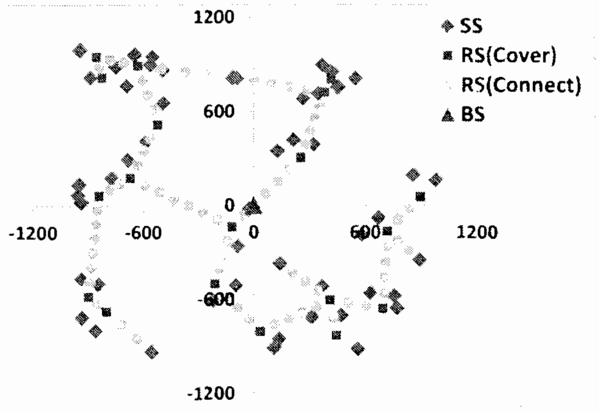


Figure 15 Tree topology (HS + MUST)

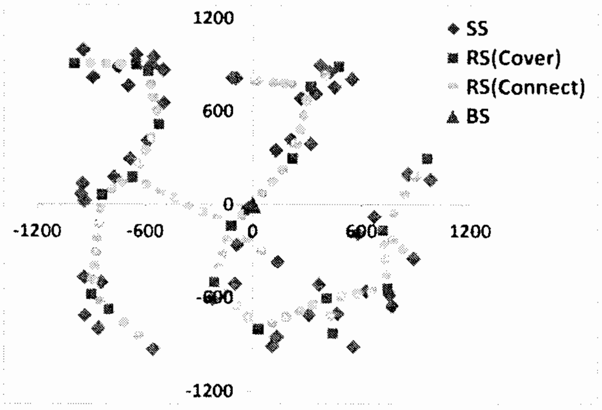


Figure 16 Tree topology (ILP + MUST)

Next, I test performances, in terms of the *number of RSs placed* and the *running times*, of our solutions. Figures 17-24 present the results using two different playing fields with different network density. In both cases, *four* metrics were tested to compare *the number of coverage RSs* (LORC), *the number of connectivity RSs* (MUST), *number of total deployed RSs* (DARP), and *running time*.

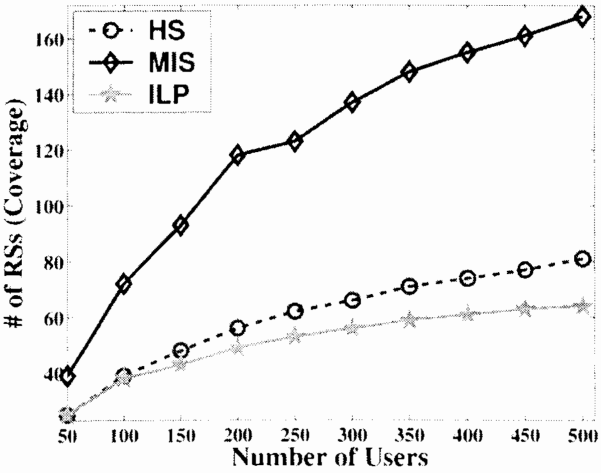


Figure 17 Coverage RSs for LORC (1000 × 1000 network)

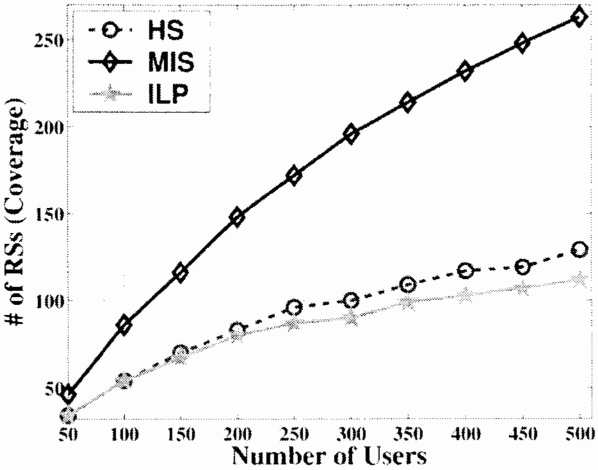


Figure 18 Coverage RSs for LORC (1500 × 1500 network)

Figures 17 and 18 showed that ILP always provide the best results for LORC, and HS provided a solution that is close to ILP and better than MIS. Meanwhile, the solutions found by MIS were always less than 3 times of the one found by ILP, which confirms our theoretical analysis.

In Figure 19, using HS as the coverage solution, I tested the performance of MUST in terms of providing connectivity RSs. Since there are no previous algorithms for MUST, and that optimal solutions are difficult to obtain, I implemented two special cases for comparison: placement with the same distance requirements d_{min} and d_{max}, respectively. The corresponding results are presented in Figure 19. As expected, our solution performs between these two special cases. I observed that the number of connectivity RSs found by MUST was less than the one with requirement d_{min}, and is *no more than 4 times* of the one found by the case with d_{max}. Similar results can be found in Figure 20.

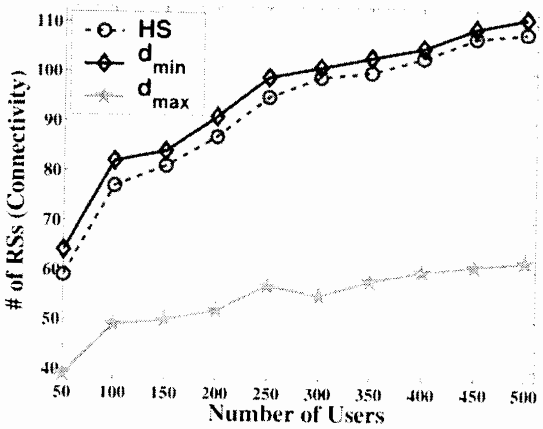


Figure 19 Connectivity RSs for LORC (1000 × 1000 network)

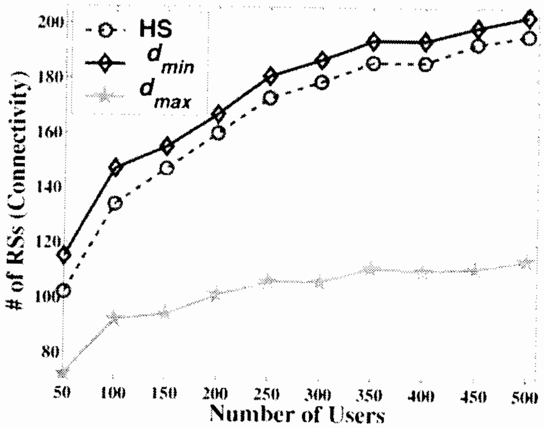


Figure 20 Connectivity RSs for LORC (1500 × 1500 network)

Figures 21 and 22 illustrated the performance of DARP, which provided the total number of RSs placed. First, I noticed that the number of RSs increased as the number of SSs increased. ILP, the best solution for LORC, seemed to provide best overall solution. And HS+MUST performed better than MIS+MUST. It seems that *the coverage RS placement has important effects on the overall placement performance.*

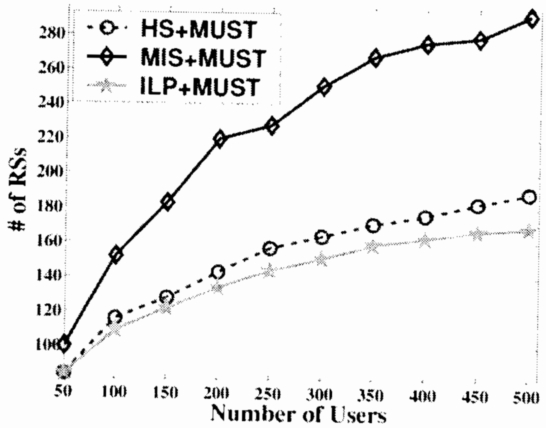


Figure 21 Overall performance for DARP (1000 × 1000 network)

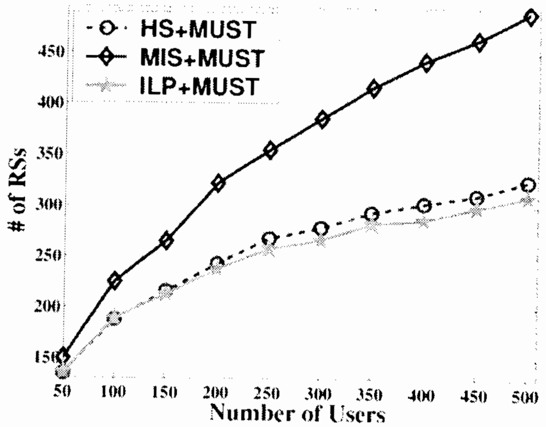


Figure 22 Overall performance for DARP (1500 × 1500 network)

Figure 23 demonstrated the running time performances. We can see that MIS had much better running time than HS and ILP, which makes it to be the best solution for large number of users. Similar trends were found in Figure 24 with different network density.

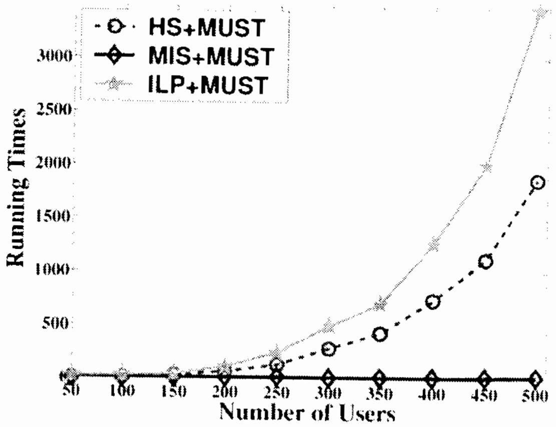


Figure 23 Running times (1000 × 1000 network)

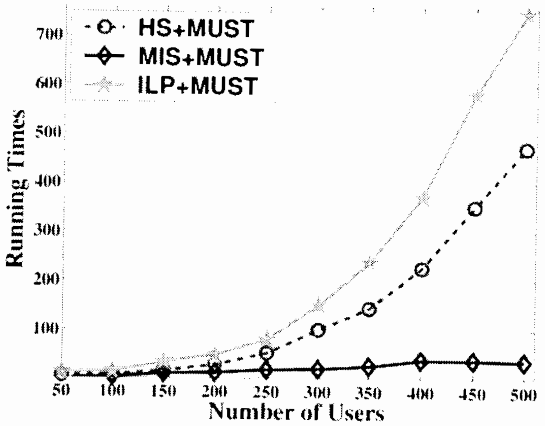


Figure 24 Running Times (1500 × 1500 network)

CHAPTER 3. MAX-MIN FAIR SCHEDULING IN OFDMA-BASED WIMAX MESH NETWORKS

For the multi-hop WiMAX mesh networks, one of the important challenges is the *multi-hop scheduling scheme* for the network. The physical layer of WiMAX uses scalable-OFDMA (Orthogonal Frequency-Division Multiple Access) since OFDM has two-fold benefits in terms of robustness to multi-path fading, and ease of digital signal processing implementation. An OFDMA system is defined as one in which each user occupies a subset of subcarriers (an OFDMA sub-channel), and *each sub-channel is assigned exclusively to one user at any time*. In OFDMA, users are not overlapped in frequency domain at any given time in one cell, which eliminates the co-channel interference in the same cell. Moreover, the frequency bands assigned to a particular user may change over time as shown in Figure 25 (each type of shade represents resources allocated exclusively to a user).

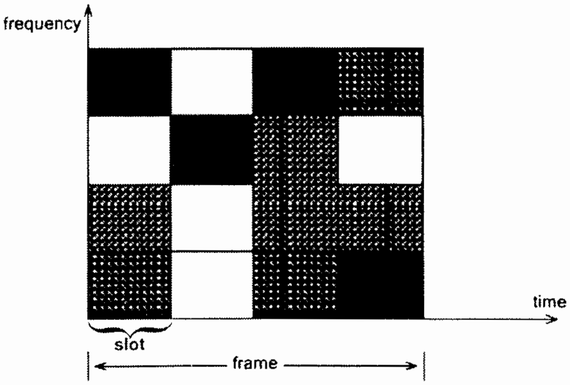


Figure 25 OFDMA in frequency and time domain

This chapter is centered on the scheduling technique for the WiMAX mesh networks. OFDMA is typically used for WiMAX network scheduling. The heart of most scheduling problems in OFDMA relay networks is assigning transmission opportunities (sub-channel, time slot) to each link in the network to maximize a certain objective function [1]. In relay networks, there are additional constraints due to synchronization in a multi-hop topology, use of a single transceiver at the relays, and flow conservation due to multi-hop relaying and fairness consideration among SSs. Another challenge is that the scheduling decisions in WiMAX networks have to be made in a timely fashion. Due to the typical order of magnitude of coherence time of the channel [34], the schedule is typically disseminated once every 5-10 ms. Thus, the problem of scheduling for fair-rate allocation in WiMAX relay networks poses several technical challenges. The objective of this chapter is to provide a comprehensive WiMAX-base network resource scheduling and allocation.

Related Works

Network resources such as channels and time slots are often limited in wireless networks. It is desirable to have a systematic scheme for fair allocation. A fundamental characteristic of a wireless network is that the channel over which communication takes place is often time-varying. IEEE 802.16j is expected to adopt Orthogonal Frequency Division Multiple Access (OFDMA) with adaptive modulation and coding at the PHY-layer.

The network scheduling and resource allocation with relay stations received much attention in recent years in the wireless networks, including WiMAX mesh networks. In [51], the authors studied scheduling with a small number of relays in cellular wireless networks and proposed a centralized downlink scheduling scheme. In [33], the authors proposed a scheme termed as OFDM2A that considers frequency-selectivity and provides significant gains over round-robin scheduling. Hierarchical architecture of mobile backbone networks are studied in [42]. Mobile backbone nodes are placed to provide communication connections for the network. Meanwhile, each regular node in the network is assigned to one backbone node. Two related problems, Maximum fair placement and assignment and maximum throughput placement and assignment, were discussed. The problem of scheduling in OFDMA-based IEEE 802.16j based WiMAX network was studied in [12]. The authors presented linear programming based heuristics for MAC scheduling in WiMAX relay networks in a fair manner while exploiting the multiuser diversity. [23] studied the capacity of the OFDMA relay networks. Two relay schemes, amplify-and-forward relay and decode-and-forward relay are analyzed. Relay node selection algorithm was presented to optimize the network capacity. [13] proposed a centralized scheduling algorithm for WiMAX mesh networks. Each node has one transceiver with multiple channels. The BS makes schedules intending to eliminate the secondary interference for reducing the length of scheduling. A resource allocation protocol that allocates subcarriers to cooperating subscriber and relay stations was proposed in [24]. [22] presented a centralized heuristic algorithm to allocate power and sub-carriers to user nodes and relays in a network where the node can establish a connection either through a direct connection or through the one relay but not in cooperative mode. But both work focused on maximizing the total network throughput rather than considering each user's required data rate. In [4], the sub-channel and relay station allocation problem was studied for the two-hop relay model. Each SS is allocated sub-channels and RSs that are required to satisfy its minimum rate requirement. A 0/1 Integer Programming was formulated with QoS and synchronization constraints. Though resource allocation has been studied for multi-cell cellular network, most related work in WiMAX network has been limited to single-cell scenarios. Applying existing resource allocation algorithm to WiMAX networks is not trivial [25]. Han *et al.* [19] proposed a distributive non-cooperative game to perform sub-channel assignment, adaptive modulation, and power control for multi-cell multiuser OFDMA networks.

Problem Statement

In this chapter, IEEE 802.16j Mobile Multi-hop Relay-based (MMR) network is used as the model for the network infrastructure. As suggested by the WiMAX standard [57], a tree rooted at the BS is usually constructed to support packet forwarding in a WiMAX mesh network. The BS is the root of the tree, the RSs are the intermediate nodes of the tree and the SSs are the leaf nodes of the tree. I focus primarily on the *scheduling for SSs and RSs over time and frequency*. I model only the uplink scenario, i.e. traffic flows from SSs to the base station. The extension to handle downlink resource allocation is along similar lines.

The IEEE 802.16 series standards [57]–[59] include the PHY and MAC layer specifications but do not specify the scheduling algorithm or the routing protocol, which are the key components for mesh networking. Previous research on adhoc and wireless mesh networking primarily concentrated on 802.11 systems which are significantly different from WiMAX systems in terms of the MAC layer scheduling scheme, interference suppression and spatial multiplexing. In this thesis, I investigate the scheduling problem in *multi-hop relay WiMAX mesh networks* with *time-varying sub-channels*. The objective is to provide a fair and efficient complete schedule to ensure the minimum satisfaction ratio among all the SSs is maximized, which has not been addressed before.

In a WiMAX network with a subscriber station set $S = \{s_1, s_2, \ldots, s_n\}$ and relay station sets $R = \{r_1, r_2, \ldots, r_m\}$. SSs share a set of sub-channels $\mathrm{H} = \{h_l \mid 1 \le l \le N_h\}$. Using two-hop relay cooperative AF protocol, the maximum achievable rate in (bits/sec/Hz) by a subscriber station s_i on subcarrier h_l with the cooperation of r_h is proved in [4], [23] to be:

$$I_l^{ih} = \frac{1}{2}\log_2(1 + \frac{\gamma_l^{id}}{N_0} + \frac{|\beta_l^h|^2 |\gamma_l^{ih}|^2 |\gamma_l^{hd}|^2}{|\beta_l^h|^2 |\gamma_l^{hd}|^2 \ N_0}) \tag{26}$$

where $|\gamma_l^{id}|^2, |\gamma_l^{ih}|^2, |\gamma_l^{hd}|^2$, respectively, are the lth subcarrier SNR from s_i to d, s_i to r_h and r_h to d. β^h is relay r_h's amplifying gain. This gives us the information that *in any time slot, the channel capacity of an SS s_i with RS r_j using sub-channel h_l*.

In each scheduling frame, BS computes and broadcasts the schedule for the entire cell. Also the channel capacities of each link at different time slots in a time frame are known to the BS at the beginning of every frame. The IEEE 802.16j standard has specified methods for this [59].

In this chapter, I study how to schedule and allocate subcarrier and time slots for each SS in a time frame. In other words, with the channel capacity of each SS given in each time slot, we need to allocate time slots and sub-channel in a frame to each user to achieve max-min fairness.

Definition 7 (Multi-hop FAir Scheduling for Throughput Optimization (MFASTO)). Based on WiMAX standard [57], a tree network G is given, with a base station BS as the root, a set of subscriber users $S = \{s_1, s_2, \ldots, s_n\}$ as the leaf nodes, a set of relay stations $R = \{r_1, r_2, \ldots, r_m\}$ as the intermediate nodes, the link capacity $c_{i,h}^l$ of each SS

s_i at time slot t using sub-channel h, and the package size requirement p_i of each SS s_i in one frame, the MFASTO problem seeks a complete schedule for a scheduling frame. Specifically, we want to find sub-channel-timeslot pair (denoted as STP in the following of this chapter) to each SS in a scheduling frame such that the minimum satisfaction ratio among all SSs is **maximized** with the following constraints:

(1) There is no spatial reuse for any pair of links which interfere with each other.

(2) An RS has only single transceiver, and cannot transmit and receive at the same time.

(3) The total data sent by an RS to BS in a frame must equal to the data it receives from its children in the frame.

Proposed Solutions

A. Integer Linear Programming for MFASTO

In [12], it was proved that scheduling with constant channel capacity is NP-hard. Therefore, our scheduling problem with time-varying channel will be NP-hard. To find an optimal solution, I provide an Integer Linear Programming (ILP) for the MFASTO problem.

I denote S, R, H and T as the set of SSs, RSs, sub-channels, and timeslots, respectively. The tree network is denoted as G. For each node $i \in S \cup R$, $pa(i)$ denotes the parent of i on the tree-topology. For each RS $r \in R$, I use $cd(r)$ to represent the set of children (SSs or RSs) of r. On the other hand, for each node i in G, $c_{t,h}^{i}$ represents the link capacity of $(i, pa(i))$ in timeslot t using sub-channel h. $f_{t,h}^{i,j}$ denotes that if the link (i, j) is assigned with time slot t and sub-channel h.

I adopt the method in [43] and [50] to identify whether two links has interference or not. Based on this method, for each node i, we can determine the set of nodes which interfere with i, denoted as $I(i)$.

$$\textbf{Maximize } \min_{i \in S} \frac{1}{p_i} \sum_{h=1}^{|H|} \sum_{t=1}^{|T|} c_{t,h}^{i} \cdot f_{t,h}^{i,pa(i)} \tag{27}$$

subject to:

$$\max_{i' \in I(i)} f_{t,h}^{i',j'} + f_{t,h}^{i,j} \leq 1, \forall h \in H, \forall t \in T, \forall i \in S \cup R \tag{28}$$

$$\max_{h \in H} f_{t,h}^{r,pa(r)} + \max_{h \in H, v \in cd(r)} f_{t,h}^{v,r} \leq 1, \forall t \in T, \forall r \in R \tag{29}$$

$$\sum_{v \in cd(r)} \sum_{h=1}^{|H|} \sum_{t=1}^{|T|} f_{t,h}^{v,r} \cdot c_{t,h}{}^{v} \leq \sum_{h=1}^{|H|} \sum_{t=1}^{|T|} f_{t,h}^{r,pa(r)} \cdot c_{t,h}^{r}, \forall r \in R \tag{30}$$

$$f_{t,h}^{i,j} = \{0,1\}, \forall (i, j) \in G, \forall t \in T, \forall h \in H \tag{31}$$

In the ILP formulation, Constraint (28), which is the Spatial Reuse Constraint, states that a particular STP can be used no more than once in each pair of interference links; Constraint (29) is the Single Transceiver Constraint which states that an RS cannot transmit and receive package concurrently due to that each RS just has one single transceiver; Constraint (30) is the Flow Constraint that all the data an RS receives in a frame must be sent out in the same frame.

B. Heuristic Algorithm

Though the ILP solution can be used to obtain optimal solutions for small sized problem, it has high time and space consumption for large-sized networks. Therefore, in practice, heuristics algorithms are needed for better running time and scalability.

For the set of SSs $S = \{s_1, s_2, \ldots, s_n\}$, we are given a corresponding set P of data package demands. Given $C = \{c_0, c_2, \ldots, c_N\}$ which is the set of capacities of all the nodes on G, where N is the total number of nodes in G. Sets T and H are the set of available timeslots and the set of available sub-channels in a frame, respectively. A collection $\mathrm{I} = \{I(v_i) \mid 1 \le i \le N\}$ can be pre-determined, where $I(v_i)$ is the set of nodes which interfere with v_i on G.

I use Algorithm 5 to allocate STPs for each node in the network. Our heuristic algorithm has the following main steps:

(1) Timeslot allocation for hops (Algorithm 6): The first step is to assign timeslots for each hop in the network. This step can guarantee that each node will not transmit and receive data package concurrently.

(2) STP allocation for nodes (Algorithm 7): The second step is to allocate STPs for each node in the network, based on the timeslots allocation results in the first step. In this step, I assume that there is no spatial reuse in the whole network. This assumption can guarantee us to obtain a resource allocation without any interference.

(3) Maximum Flow Improvement (Algorithm 8): After allocating STPs for each node, I use a maximum flow based algorithm to improve the network throughput.

Algorithm 5 Schedule (G, P, C, T, H, I)

```
Construct a set Q $= \{q_0, \ldots, q_N\}$ which is the set of STP demands of all the nodes on $G$
Construct $\mathrm{L}^{\mathrm{D}} = \{l_k^D \mid 1 \le k \le K\}$ and $\mathrm{L}^{\mathrm{A}} = \{l_k^A \mid 1 \le k \le K\}$, where $l_k^D$ is the timeslot
  demand of Hop $k$, and $l_k^A$ is the number of timeslots allocated to Hop $k$; Initialize each
  $l_k^D$ and $l_k^A$ to 0;
Construct a set $\Lambda = \{\lambda_i \mid 0 \le i \le N\}$, where $\lambda_i$ is the size of package node $v_i$ have to send
  to its parent; Construct a set of $\Delta = \{\delta_i \mid 0 \le i \le N\}$, where $\delta_i$ is the number of STPs node
  $v_i$ needs; Initialize each $\lambda_i$ and $\delta_i$ to 0;
for each node $v_i$ on $G$ do
    $p_i \leftarrow \sum_{v_j \in G_i} p_j$; $q_i \leftarrow \left\lceil \frac{p_i}{c_i} \right\rceil$; ($G_i$ is the sub-tree rooted at $v_i$)
end for
for $k \in \{1, 2, \ldots, K\}$ do
    $l_k^D \leftarrow \left\lceil \frac{\sum_{v_i \in V_k} q_i}{|H|} \right\rceil$; //$V_k$ is the set of nodes $k$ hops from BS
end for
T $\leftarrow$ Hops Allocation ($T$, $\mathrm{L}^{\mathrm{D}}$, $\mathrm{L}^{\mathrm{A}}$);
Construct a collection X $= \{X_k \mid 1 \le k \le K\}$;
for $k \in \{1, 2, \ldots, K\}$ do
    Construct a set of STPs $X_k \leftarrow H \times T_k$;
end for
Total number of STPs $\tau \leftarrow |H||T|$;
U $\leftarrow$ Nodes Allocation ($\tau$, $\Lambda$, $\Delta$, Q, X);
U$'$ $\leftarrow$ Max Flow Improvement ($\Lambda$, $\Delta$, I, U);
return U$'$
```

Algorithm 6 Hop Allocation (T, L^{D}, L^{A})

```
$\gamma \leftarrow \frac{|T|}{\sum_{k=0}^{K} l_k^D}$;
for all $k \in \{1,2,\ldots,K\}$ do
    $l_k^A \leftarrow \lceil \gamma l_k^D \rceil$; $\gamma_k \leftarrow l_k^A / l_k^D$; $\gamma_k' \leftarrow (l_k^A - 1) / l_k^D$. ( $\gamma_k \leftarrow 1$ and $\gamma_k' \leftarrow 1$ if $l_k^D = 0$ )
    // $\gamma_k'$ is the satisfaction ratio if subtract 1 timeslot from $l_k^A$
end for
$L_{over} \leftarrow \sum_{k=1}^{K} l_k^A - |T|$;
while $L_{over} > 0$ do
    Choose a hop $j$ with the greatest $\gamma'$;
    $l_j^A \leftarrow l_j^A - 1; \gamma_j \leftarrow l_j^A / l_j^D; \gamma_j' \leftarrow (l_j^A - 1) / l_j^D$;
    $L_{over} \leftarrow L_{over} - 1$;
end while
for all $k \in \{1,2,\ldots,K\}$ do
    Construct set $T_k$ of timeslots; Choose $l_k^A$ elements from set $T$ and add them to $T_k$;
    $T \leftarrow T \setminus T_k$;
end for
return $\mathrm{T} = \{T_k \mid 1 \le k \le K\}$;
```

Algorithm 7 Nodes Allocation (τ , Λ , Δ , Q , X)

```
$\theta \leftarrow \tau / \sum_{v_i \in G} q_i$ ( $\theta \leftarrow 1$ if $\theta > 1$ );
for all SS node $v_i \in G$ do
    $\delta_i \leftarrow \lceil \theta q_i \rceil; \lambda_i \leftarrow \delta_i c_i$; ( $\lambda_i \leftarrow p_i$ and $\delta_i \leftarrow \left\lceil \frac{p_i}{c_i} \right\rceil$ if $\lambda_i > p_i$ )
    $s_i \leftarrow \frac{\lambda_i}{p_i}; s_i' \leftarrow \frac{(\delta_i - 1)c_i}{p_i}$; // $s_i'$ is the satisfaction ratio if subtract 1 timeslot from $\delta_i$
    for all RS $v_j$ on the path from $v_i$ to BS do
        $\lambda_j \leftarrow \lambda_j + \lambda_i; \delta_j \leftarrow \left\lceil \frac{\lambda_j}{c_j} \right\rceil$;
    end for
end for
while $\exists k, 1 \le k \le K$ and $| X_k | < \sum_{v_i \in V_k} \delta_i$ ( $V_k$ is the set of nodes which has $k$ hops form
  BS) do
    Choose an SS $v_i$ with the greatest $s'$;
    $\delta_i \leftarrow \delta_i - 1; \lambda_i \leftarrow \delta_i c_i; s_i \leftarrow \frac{\lambda_i}{p_i}; s_i' \leftarrow \frac{(\delta_i - 1)c_i}{p_i}$;
    for all RS $v_j$ on the path from $v_i$ to BS do
        $\lambda_j \leftarrow \lambda_j - c_i; \delta_j \leftarrow \left\lceil \frac{\lambda_j}{c_j} \right\rceil$;
    end for
end while
for all link $(i, pa(i)) \in G$ do
    Choose $\delta_i$ STPs from $X_{k_i}$ and add them to $U_i$; { $k_i$ is the number of hops from BS
      to $v_i$ }
end for
return $\mathrm{U} = \{U_i \mid v_i \in G\}$;
```

Algorithm 8 Max Flow Improvement (Λ , Δ , I , U)

```
Construct a directed graph $G_A(V,E)$ and a virtual node $s$ ; $V \leftarrow V \cup \{s\}$ ;
for all SS node $v_i \in G$ do
    Construct a set $U_i'$ of STPs;
end for
Set COUNT = 0;
while COUNT $<|V|-2$ (no allocations for BS and $s$ ) do
    Choose node $v_{min}$ with the smallest satisfaction ratio $s_{min}$ ;
    if $(U \setminus I_{min}) \neq \varnothing$ then
        Choose $(h,t) \in (U \setminus I_{min}); I_{min} \leftarrow I_{min} \cup \{(h,t)\}$ ;
        if $t$ is not used by $pa(v_{min})$ or any $v' \in cd(v_{min})$ then
            $\lambda_{min} \leftarrow \lambda_{min} + 1; s_{min} \leftarrow \frac{\lambda_{min} c_{min}}{p_{min}}$ ;
            $U'_{min} \leftarrow U'_{min} \cup \{(h,t)\}$ ;
        end if
    else
        COUNT $\leftarrow$ COUNT+1;
    end if // $I_{min}$ is the set of STPs which conflict the nodes in $I(v_{min}) \cup \{v_{min}\}$ .
end while
for all link $(i, pa(i)) \in E$ do
    Set the capacity of $(i, pa(i))$ to be $\lambda_i c_i$ ;
end for
for all leaf node $l \in G_A(V,E)$ do
    Construct a link $(s,l)$ with capacity $+\infty$ ; $E \leftarrow E \cup \{(s,l)\}$ ;
end for
Find the maximum flow from $s$ to BS and the link flows;
for all link $(i, pa(i)) \in G$ do
    if $v_i$ is an SS then
```

```
27:             $\lambda_i \leftarrow \left\lfloor \frac{f_i}{c_i} \right\rfloor$, where $f_i$ is the flow value of $(i, pa(i))$;
28:      else
29:             $\lambda_i \leftarrow \left\lceil \frac{f_i}{c_i} \right\rceil$;
30:      end if
31: Keep $\lambda_i$ elements in $U'_i$ and remove the rest;
32: return $\mathrm{U} = \{U'_i \mid v_i \in G\}$;
```

Let us use an example to illustrate our proposed algorithm from Figure 26 to Figure 30. The network topology shown in Figure 26, includes 1 BS, 2 RSs (R_1 and R_2) and 4 SSs (S_1 , S_2 , S_3 and S_4). The package requirements of these four SSs are 2, 3, 2 and 1, respectively. For simplicity, the capacity of each link is set to be 1. The number of timeslots and number of sub-channels in a frame are 3 and 2. The interference node sets of all nodes are: $I(R_1)=\{R_2,S_1,S_2\}$, $I(R_2)=\{R_1,R_3,R_4\}$, $I(S_1)=\{R_1,S_2\}$, $I(S_2)=\{R_1,S_1,S_3\}$, $I(S_3)=\{R_2,S_2,S_4\}$ and $I(S_4)=\{R_2,S_3\}$.

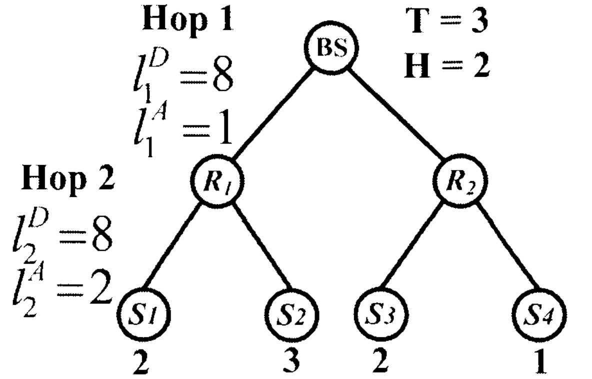


Figure 26 Network topology

Given the network topology in Figure 26, we first calculate the timeslot demand for each hop (Lines 1 - 6 in Algorithm 5). Consequently, the timeslot demands are obtained, which is 8 for both Hop 1 and Hop 2. Then, we call Algorithm 6 (Line 10) to allocate timeslots for each hop.

In Algorithm 6, the ratio γ (the total number of timeslots divided by number of total timeslot demands in this network) is calculated, which is $\frac{3}{16}$ in this case (Line 1). After that, we pre-allocate timeslots for each hop based on γ (Lines 2- 4). The corresponding results $l_1^A = 2, \gamma_1 = \frac{1}{4}, \gamma_1' = \frac{1}{8}$ and $l_2^A = 2, \gamma_2 = \frac{1}{4}, \gamma_2' = \frac{1}{8}$ are obtained. Consequently, the over-allocated timeslots $L_{over} = 1$, which means we have to subtract 1 timeslot from one of the hops (Line 5). From Line 6 to Line 10, we choose one hop with the maximum γ' and subtract 1 timeslot from this hop. We repeat the same procedure until L_{over} becomes 0. As a result, we allocate 1 timeslot to Hop 1, and allocate 2 timeslots to Hop 2 (Lines 11-14). Therefore, we assign Timeslot 2 to Hop 1, and assign Timeslots 0 and 1 to Hop 2. The allocation results are shown in Figure 26.

Then, back to Algorithm 5, we allocate STPs for each node (Lines 11-15). At this time, we assume that, for any sub-channel, there is no spatial reuse in the whole network. This assumption can guarantee that the potential interference links cannot interfere with each other. From Line 11 to Line 14, we constructed set of STPs for each hop. More specifically, $X_1 = \{(h_0, t_2), (h_1, t_2)\}$ and $X_2 = \{(h_0, t_0), (h_1, t_0), (h_0, t_1), (h_1, t_1)\}$. At Line 15, the total number of available STPs is calculated, which is $\tau = 2 \times 3 = 6$ in this case. Then, we call Algorithm 7 to allocate resources for all the nodes in this network (Line 16) shown in Figure 27.

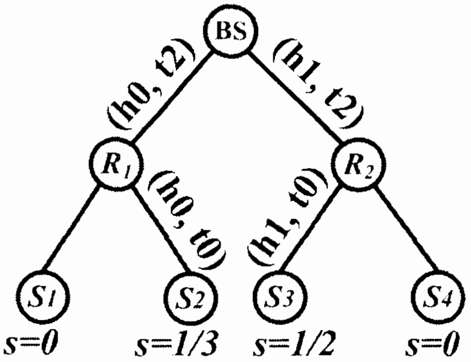


Figure 27 Scheduling assuming no spatial reuse in network

In Algorithm 7, we use the similar idea of Algorithm 6. We first calculate a ratio $\theta = \frac{6}{2+3+2+1+5+3} = \frac{3}{8}$ (Line 1). Then, we allocate STPs using this ratio (Lines 2 - 8). For example, we assign $\lambda_1 = \lceil \theta \cdot q_1 \rceil = \left\lceil \frac{3}{8} \cdot 2 \right\rceil = 1$ STP for S_1 (Lines 3 - 4). After that, we update the assigned STPs of the nodes on the path from BS to S_1 (Lines 5 - 7). After allocating resources for all the nodes, we subtract the over-allocated STPs from some nodes. We choose the SS with greatest s' in the network, and subtract 1 STP from this node (Lines 10 - 11). Then, we update the allocated STPs for the nodes on the path from BS to it (Lines 12 - 14). In this case, we subtract I STP from S_1 and S_4 , and update the STPs of their parent nodes. The corresponding allocation results, shown in Figure 27, are returned to Algorithm 5.

Then, Algorithm 8 is called to allocate resources allowing spatial reuse if no interference (Line 17 in Algorithm 5). In Algorithm 8, we choose an SS with smallest satisfaction ratio (Line 7 in Algorithm 8), and check whether there is any STP can be used for it (no interference and no time confliction) (Lines 8 - 16). If possible, we allocate 1 STP for this node (Lines 10 - 13). This procedure is repeated until there is no more available

STP for any SS. As the result, we allocate 1 STP for S_1, S_4, R_1 and R_2, shown in Figure 28.

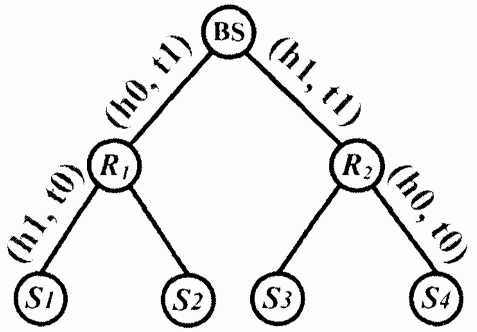


Figure 28 Scheduling with spatial reuse if no interference

Then, for each link $(i, pa(i))$, whose source node is not the virtual source node s, in the constructed auxiliary graph, we assign its capacity with $c_i \cdot \lambda_i$ (Lines 18 - 20). For the link whose source node is s, we assign its capacity with $+\infty$ (Lines 21 - 23). The auxiliary graph and the link capacities are shown in Figure 29.

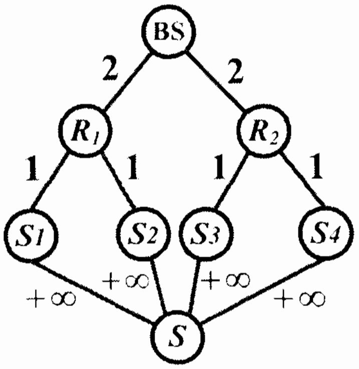


Figure 29 Maximum flow

After that, we calculate a maximum flow in this graph, and output the corresponding STP allocation (Lines 24 - 32). The final resource allocation results are shown in Figure 30. The minimum satisfaction ratio in this case is $\frac{1}{3}$, and the network throughput is 4.

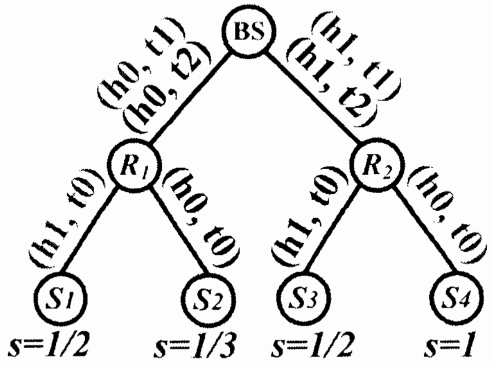


Figure 30 Result: throughput = 4, *Smin* = 1/3

Numerical Results

In this section, I presented numerical results to evaluate the performances of our solutions. I implemented the ILP solution and our proposed heuristic algorithm. To evaluate our heuristic algorithm, I divided it into to sub-solutions, the algorithm with Maximum Flow improvement and the one without Maximum Flow improvement, which were denoted as MaxFlow and NoResue in the figures.

All our simulation runs were performed on a 2.8 GHz Linux PC with 2G bytes of memory. I used different network topologies in different playing fields (1500×1500 sq. units, 3000×3000 sq. units and 4000×4000 sq. units) to evaluate our proposed solutions. The transmission range and interference range of each SS were set to be 500 and 1000, respectively. For RS and BS, the transmission range and interference range were 1000 and 2000. One base station was deployed at the center of the field. For the 1500×1500, 3000×3000 and 4000×4000 playing fields, I distributed 4, 16 and 36 RSs uniformly to cover the whole network area. Multi-hop shortest path routing was adopted to obtain the network topology. The SSs were distributed randomly and uniformly in the playing fields. The data package requirement of each SS and the link capacity between it and its parent were randomly distributed in [2, 8] and [5, 10], respectively. All the figures illustrated the average of 10 test runs for various scenarios.

First, I compared the minimum satisfaction ratios and the running times obtained by the ILP formulation and our heuristic algorithm in a 1500×1500 sq. units playing field with 4 RSs. I used Gurobi Optimizer [61] to solve the ILP formulation. Due to the limitation of memory space, I set the number of timeslots and number of sub-channels in a frame to 12 and 5, respectively. The number of SSs was increased from 5 to 25. The corresponding results were shown in Figure 31 and Figure 32. In Figure 31, I noticed that when the number of SSs was more than 15, the ILP formulation cannot provide solution due to the memory limitation. On the other hand, comparing with optimal solution, a good performance can be obtained from our heuristic algorithm. More specifically, when the number of SSs was no more than 10, both NoReuse and MaxFlow can achieve a minimum satisfaction ratio 1, as the one delivered by ILP formulation. When the number of SSs was 15, the minimum satisfaction ratio of optimal solution is 0.969, while the ones of NoReuse and MaxFlow are 0.949 and 0.926, which were close to the optimal solution. In Figure 32, I tested the running time performances of ILP formulation and our heuristic. The metric for ILP formulation was seconds, while the metrics for NoReuse and MaxFlow were milliseconds. From Figure 32, it is easy to see that our heuristic algorithm is much faster than the ILP solution.

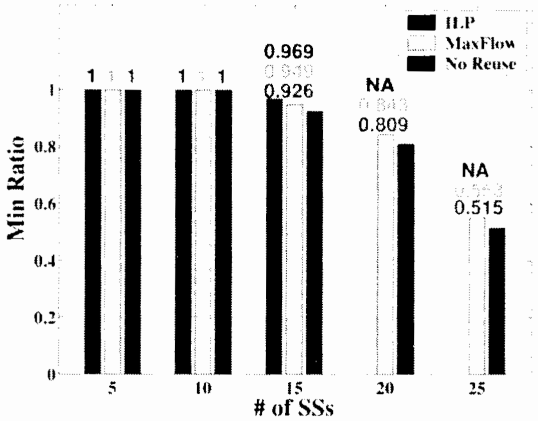


Figure 31 Minimum satisfaction ratio

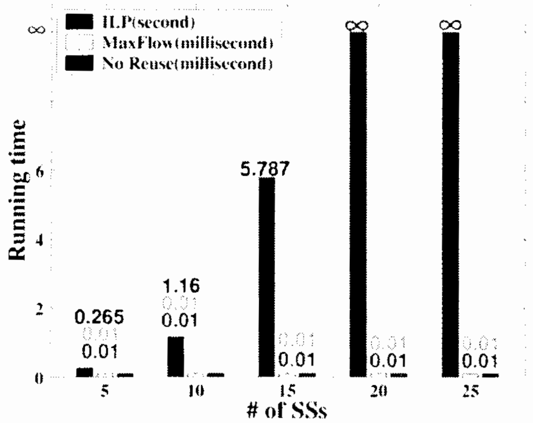


Figure 32 Running time

Then I tested the performances of NoReuse and MaxFlow in terms of minimum satisfaction ratio, average satisfaction ratio and network throughput in larger playing field with more SSs. In this network, I set the number of timeslots and number of sub-channels in a frame to 48 and 5. The number of SSs was increased from 10 to 100. Figures 33-38 presented the results using different playing fields (3000×3000 sq. units and 4000×4000 sq. units) and different numbers of RSs (16 and 36).

In Figure 33, when the number of SSs was no greater than 30, the minimum satisfaction ratios achieved from both NoReuse and MaxFlow among all the SSs were 1. As the number of SSs increased, the total package requirements also increased.

Consequently, the minimum satisfaction ratio decreased. From Figure 33, I also observed that the minimum satisfaction ratios obtained from NoReuse and MaxFlow were similar in this network topology. In Figure 34, the minimum satisfaction ratio decreased more sharply due to the increased number of hops and RSs.

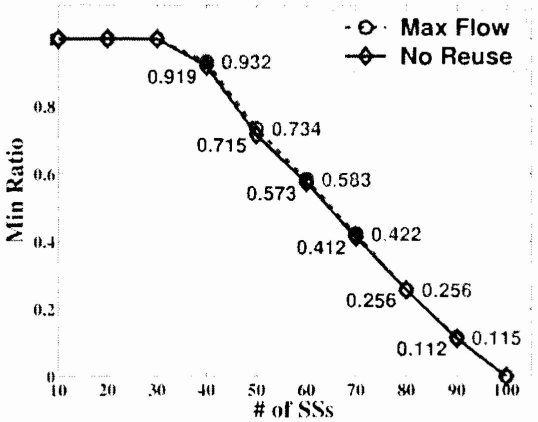


Figure 33 Minimum satisfaction ratio in 3000 × 3000 playing field with 16 RSs

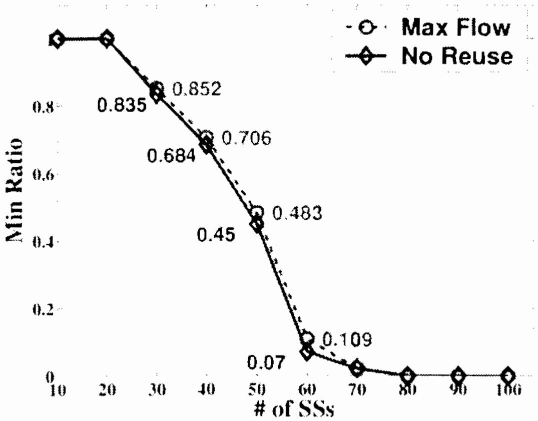


Figure 34 Minimum satisfaction ratio in 4000 × 4000 playing field with 36 RSs

As shown in Figure 35, as the number of SSs increased, the average satisfaction ratios decreased. Figure 35 also showed the average satisfaction ratio from MaxFlow was better than the one from NoReuse. More specifically, MaxFlow can improve the average satisfaction ratio of NoReuse by up to 6%. In Figure 36, at each point, the difference between the performances of MaxFlow and NoReuse were much greater than the one in

Figure 36. In Figure 36, MaxFlow can improve the average satisfaction ratio of NoReuse by up to 7%. This is because, after NoReuse, the number of potential available resources in large network was more than the one in a relative small network.

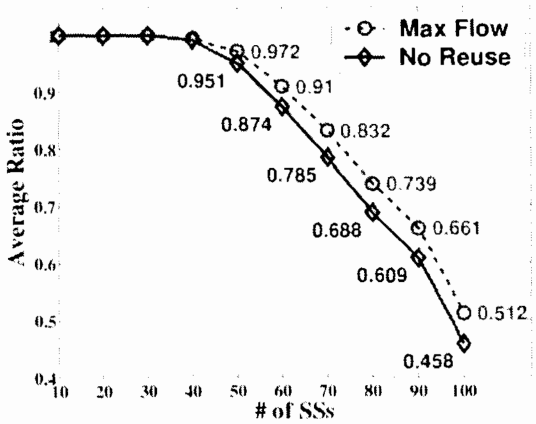


Figure 35 Average satisfaction ratio in 3000 × 3000 playing field with 16 RSs

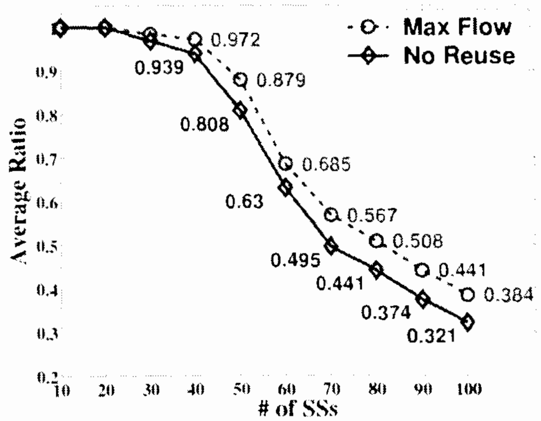


Figure 36 Average satisfaction ratio in 4000 × 4000 playing field with 36 RSs

In Figure 37, the performance of network throughput of MaxFlow was better than the one obtained from NoReuse. Also, when the number of SSs was no greater than 70, the network throughput performances of NoReuse and MaxFlow increased as the the network size increased. However, the network throughputs decreased when the number of SSs was greater than 70. Because of the fixed number of resources (timeslots and sub-channels), in order to obtain a higher minimum satisfaction ratio, more resources have to be allocated to

RSs to forward packages for SSs. Therefore, the network throughputs decreased after when the network size reach a "threshold" value. Similar trends were found in Figure 38.

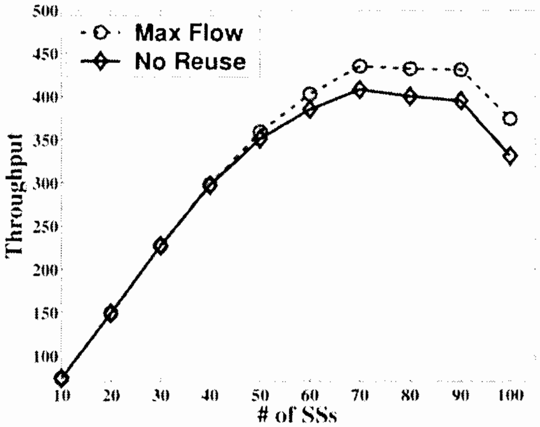


Figure 37 Network throughput in 3000 × 3000 playing field with 16 RSs

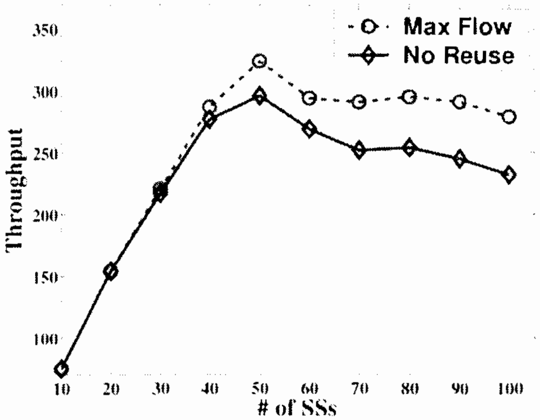


Figure 38 Network throughput in 4000 × 4000 playing field with 36 RSs

CHAPTER 4. COGNITIVE RADIO SCHEDULING FOR OVERWATER COMMUNICATIONS

While extensive research has been carried out examining the effects of terrain and mobility on wireless communications in different network topologies, the unique effects of propagation over water and their impact on wireless networking have not been well studied. In this chapter, I study wireless communications over water, which may suffer from serious multipath fading due to strong specular reflections from conducting water surfaces.

Overwater propagation is a special case of the general ground reflection problem. The large scale fading characteristics for a link whose transmitting and receiving nodes are close to the ground are well captured by the two-ray model, leading to the well-known d^{-4} path loss formula [39], where d is the distance between transmitting and receiving nodes. In the case where the E-field is in the plane of incidence and the surface is a strong reflector, the exact expression for the received power P is given by Equation 1 [39].

$$P(d) = \frac{Q}{d^2} \sin^2(\frac{\theta_\Delta}{2}) \tag{32}$$

In this equation, θ_Δ is the phase difference between direct and reflected signals, which is related to antenna heights, distance and operating frequency. Q is a constant. This two-ray effect can lead to deep fades under conditions when $\theta_\Delta = k\pi$ (null conditions), where k is an integer. Once d is sufficiently large, and the power then falls off asymptotically with the increasing distance.

Figure 39 and Figure 40 show an example of the two-ray effect and overwater path losses predicted by the Advanced Refractive Effects Prediction System (AREPS) [62]. The power loss of a path over ocean on two different operating frequencies, 2.4GHz and 1.7GHz, as a function of distance between transmitting and receiving nodes. The heights of both transmitting and receiving antennas are 60m. The power loss predicted by the AREPS (solid black line) oscillates about the large scale free space power loss (the red line), with extremes ranging up to 30dB.

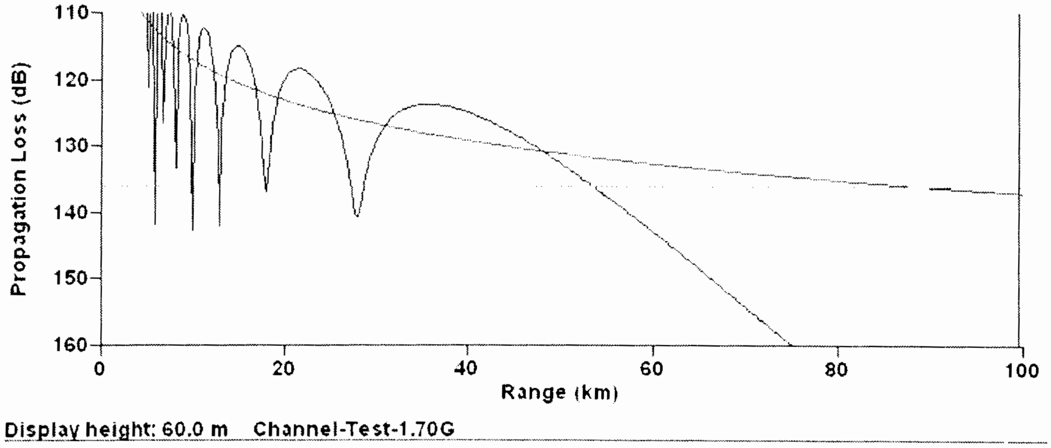


Figure 39 Overwater path loss on 1.7 GHz given by AREPS

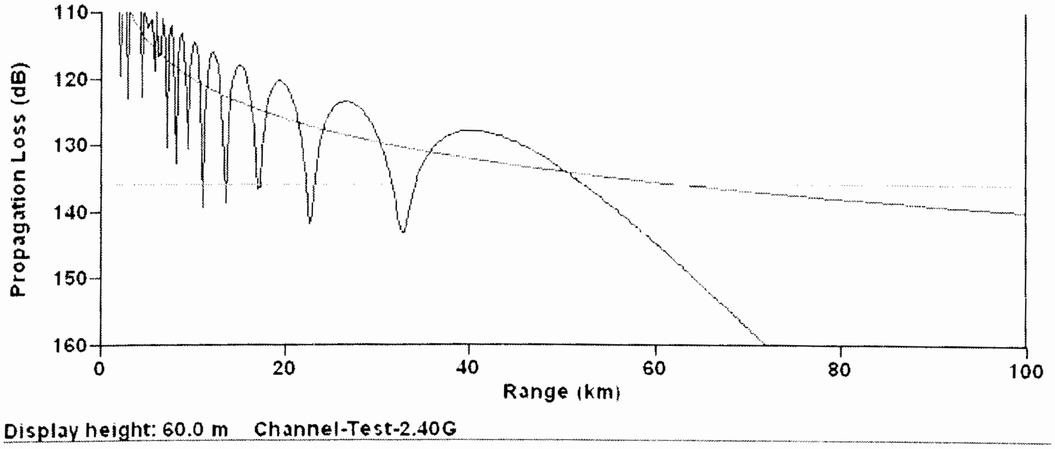


Figure 40 Overwater path loss on 2.4 GHz given by AREPS

This effect can be avoided by a change in frequency, i.e., using a different channel. The cognitive radio technology enables dynamic spectrum access [2]. With a cognitive radio, an MS can dynamically switch its radio to any available channel when it has packets to send. In this chapter, I formally define the related problem as the OVErwater Radio-Time Scheduling (OVERTS) problem which seeks a channel-time assignment schedule such that a "good" communication link can be maintained between each MS and the BS satisfying the time-slot requirement of the particular MS.

Related Works

Overwater path loss effects have generally been ignored until recently, as the focus of attention in wireless system design and applications has been toward cellular systems and wireless LANs. Empirical evidence of this effect has recently been reported for an overwater LOS path in [6].

Spectrum allocation and scheduling are very important problems in cognitive radio networks [2]. In a centralized spectrum sharing protocol in [7], spectrum management is conducted in a central server, which can obtain a global view of network by exchanging information with users. In [8], the authors presented a distributed spectrum allocation algorithm based on local bargaining. In [56], Zhao *et al.* presented optimal and suboptimal distributed spectrum access strategies under a framework of partially observable Markov decision process. In [53], the authors introduced the concept of time-spectrum block and proposed algorithms to allocate such blocks to satisfy particular performance goals. An effective heuristic algorithm for the scheduling problem in wireless backhaul networks with smart antennas was presented in [50]. In [46], Tang *et al.* studied joint channel allocation and scheduling problems in cognitive radio networks. Optimal and heuristic algorithms were presented to find maximum throughput and fair solutions.

Channel assignment has also been studied for traditional wireless networks with multiple homogeneous channels. In [38] and [37], Raniwala *et al.* proposed one of the first IEEE 802.11-based multi-radio mesh network architectures and developed several centralized and distributed heuristic algorithms for channel assignment and routing. In [54], the authors proposed an effective heuristic to find a robust topology with the minimum network interference. In [47], Tang *et al.* proposed an interference-aware channel assignment algorithm. A constant bound approximation algorithm was proposed in [3] to compute channel assignment, routing and scheduling solutions for fair rate allocation.

In this work, I study channel assignment and scheduling in cognitive radio networks and overwater communications. Our problem is different from those in the related works.

Problem Statement

In this section, I will describe the system model and formally define the optimization problem.

I consider a wireless network over water, consisting of a Base Station (BS) and a set of MSs $M=\{m_1, m_2, \dots, m_n\}$, each of which is equipped with a cognitive radio. The available spectrum is divided into H non-overlapping radio channels. A cognitive radio can be tuned to an available channel to deliver its packets. A radio used by MS can usually transmit packets over a long distance with the help of a powerful amplifier. Hence, *each MS can directly communicate with the BS*. It is also assumed that each radio transmits at the fixed power level. Therefore, in such a network, *there are n MS-BS links and every MS/link needs to be assigned a different channel for communications at any time to prevent co-channel interference.*

Link capacity is related to path loss and other parameters such as transmit power, antenna gains, channel bandwidth, and so on. Once the values of the other parameters are fixed, link capacity becomes a function of path loss. In Figure 39, the orange horizontal line in the figure indicates a threshold of 138dB, corresponding to a path loss that would limit the radio link capacity to a certain acceptable level. As a function of distance between transmitting and receiving nodes, there are intervals where the path loss exceeds this threshold for a particular operating frequency. Intuitively, a channel assignment and scheduling method could be used to switch the radio to a different "good" channel whenever this happens.

For all the MSs, I use a set $D=\{d_1,d_2,\ldots,d_n\}$ to represent the time-slots demands of them, where di is MS m_i's the requirement of total time-slots for successful transmission. Each MS is assumed to know its moving trajectory and speed in the next τ seconds. Therefore, the distance between the BS and an MS at any time can be computed in advance. The BS gathers such information periodically from each MS. I define a *schedule assignment* for an MS m_i as (i,h,t_j,t_k), where $h\in\{1,2,\ldots,H\},0\le t_j\le t_k\le\tau$ which specifies a channel h and a time interval (t_j,t_k) are assigned to m_i.

Definition 8 (Feasible schedule): Given a radio-time assignment (i,h,t_j,t_k), if during the time interval $[t_j,t_k]$, the link capacity of channel h between the MS m_i and the

BS is not smaller than the capacity threshold C (according to the path loss values predicted by the AREPS), we say such a schedule assignment is a *feasible assignment*. Furthermore, a set $A_i = \{(i,h,t_j,t_k) \mid 0 \le t_j \le t_k \le \tau\}$, where $i \in \{1,2,\ldots,n\}$ and $h \in \{1,2,\ldots,H\}$, is a *feasible schedule* for m_i if each assignment in the set is feasible.

Following the scheduling of each MS, a channel and time assignment schedule for the network is given by $\vec{A} = \{A_1, A_2, \ldots, A_n\}$.

Definition 9 (Conflicted schedules): We say two assignment (i,h,t_j,t_k) and (i',h',t_j',t_k') conflict with each other if $i = i'$ and $(t_j,t_k) \cap (t_j',t_k') \ne \varnothing$, or $h = h'$ and $(t_j,t_k) \cap (t_j',t_k') \ne \varnothing$. Given any two schedule sets A_i and $A_{i'}$, if any assignment from $A_{i'}$ conflicts with an assignment from $A_{i'}$, we say A_i and $A_{i'}$ are *conflicted schedules*.

Definition 10 (OVErwater Radio-Time Scheduling (OVERTS)): Given a cognitive radio network over water with a BS, capacity threshold of C, H channels, and n MSs each with a time slot request d_i, the OVErwater Radio-Time Scheduling (OVERTS) problem seeks a radio-time schedule for the network such that during the period $[0,\tau]$, the sum of time slots assigned for each MS m_i is no smaller than d_i, and no MS has conflicted schedules.

Proposed Scheduling Algorithms

As mentioned before, the distance between the BS and an MS m_i is pre-computed. Consequently, the feasible schedule for m_i can be determined in advance. Note that usually the time intervals in a feasible schedule may not be continuous. In Figure 44, I give a sample of feasible schedules for two MSs in a time span $[0,500]$. I will use this example to illustrate our solutions throughout this chapter. In this section, I propose two effective and efficient solutions the OVERTS problem. First, I transfer the scheduling problem into a classic *maximum independent set* problem. In the second solution, I apply the dynamic programming scheme to solve the OVERTS problem.

A. Maximum Independent Set Based Algorithm

In this subsection, I propose a solution based on the *Maximum Independent Set* scheme. To seek the best schedule for all the MSs, I aim to find the maximum (number of time slots) independent (conflict-free) set of assignments for each MS. Given an auxiliary graph $G(V,E)$, a set $T=\{(i,h,t_j,t_k)\mid 0\le t_j\le t_k\le\tau, i\in\{1,2,\ldots,n\}, h\in\{1,2,\ldots,H\}\}$ of available time intervals, and the set of timeslot demands of MSs $D=\{d_1,d_2,\ldots,d_n\}$, Algorithm 9 is listed to calculate a feasible solution to OVERTS.

Algorithm 9 Graph-MIS (M , H , T)

```
Construct an auxiliary graph $G(V,E)$; $V \leftarrow \varnothing$; $E \leftarrow \varnothing$;
for each feasible assignment $(i,h,t_j,t_k)$ do
    Add the corresponding node $v$ into $G$;
end for
for each conflict schedule pair $(i',h',t'_j,t'_k)$ and $(i,h,t_j,t_k)$ do
    Add an edge between the corresponding nodes in $G$;
end for
for each edge $e \in G$ do
    if $e$ is an edge between two assignments from the same user do
        Hide edge $e$;
    end if
```

```
12: end for
13: Finding-MIS ($G$ , $D$);
14: for each MS $m_i \in M$ do
15:     Hide the nodes which is not related with $m_i$ and their edges in $G$ to form a
        sub-graph $G_i(V_i, E_i)$;
16:     $A_i \leftarrow$ Finding-MIS ($G_i$, $d_i$);
17: end for
18: return $\vec{A} = \{A_i \mid i \in \{1, 2, \ldots, n\}\}$
```

Algorithm 10 Finding-MIS (G , D)

```
1: while $E \neq \varnothing$ do
2:     for each node $v \in V$ do
3:         Node weight $w_v \leftarrow \frac{\Delta_v d_i}{t_k - t_j}$, where $\Delta_v$ represents the nodal degree of $v$;
4:     end for
5:     Find the largest weight $w_{max}$ in $V$ ;
6:     Remove $v_{max}$ from $V$ and add it into a set $R$ ;
7:     Remove $v_{max}$'s adjacent edges from $G$ ;
8: end while
9: for each node $v \in R$ do
10:    if there is no conflict between $v$ and any node left in $V$ then
11:        Add $v$ back to $V$ ;
12:    end if
13: end for
14: return the set of assignments corresponding to $V$ ;
```

We first construct a graph $G(V, E)$ to represent the assignments and confliction relations of the MSs to assist such a computation (Line 1 - Line 8). For each feasible assignment (i,h,t_j,t_k) of MS m_i, a node v is added into graph G. For any two assignment (i,h,t_j,t_k) and (i',h',t_j',t_k'), if they conflict with each other, then there is an edge between their corresponding nodes in G. After constructing the graph, we first aim to calculate a maximum independent set of the graph considering only the conflicts from different MSs (Lines 8- 13). After that, we further refine the schedule by considering the conflicts of the assignments from the same MS (Line 14- Line 17). In Line 13 and Line 16, Algorithm 10 is called to seek a maximum independent set from a graph.

Let us use the example from Figure 44 to demonstrate Algorithm 9. Based on the information in Figure 44, an auxiliary graph is constructed in Figure 41. First, after hiding the solid lines (Lines 8-12), we apply Algorithm 10 to seek a maximum independent set (MIS). The time interval information $(1,1,20,80)$, $(1,1,140,200)$, $(1,2,180,260)$, $(2,2,360,440)$, $(2,2,30,120)$ and $(2,1,300,420)$ will removed from $G(V,E)$ in sequence (Algorithm 10, Lines 1-8). The remaining time-channel graph $G(V,E)$ is showed in Figure 42.

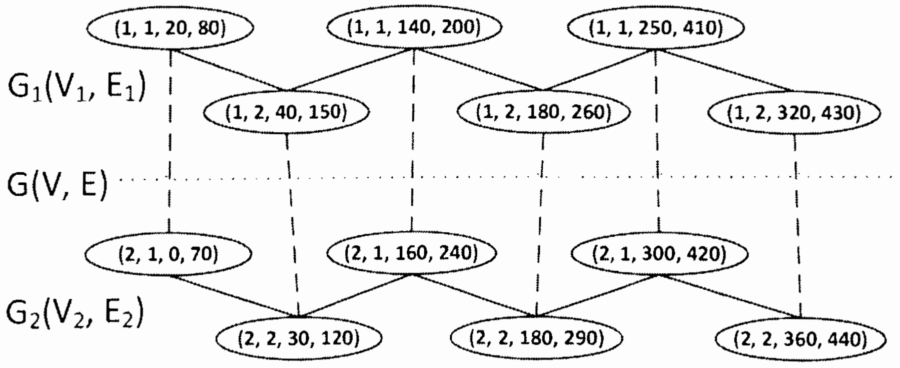


Figure 41 A time-channel graph

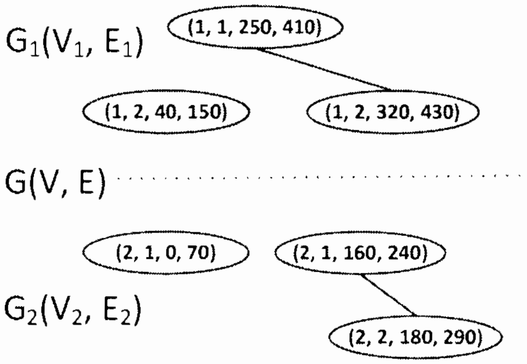


Figure 42 Remaining time-channel graph

Then, we refine the MIS considering only the solid links (Algorithm 9, Lines 14-17). Assignments $(1,2,320,430)$ and $(2,1,160,240)$ will be removed. Finally, a feasible schedule $A=\{\{(1,2,40,150),(1,1,250,410)\},\{(2,1,0,70),(2,2,180,290)\}\}$ is shown in Figure 43.

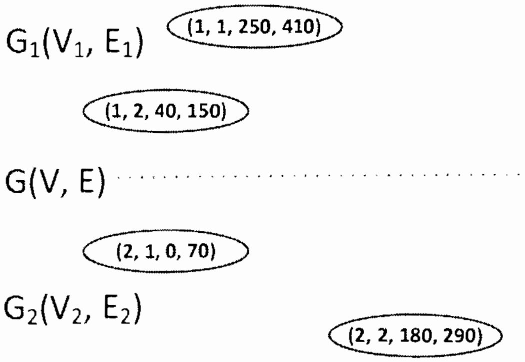


Figure 43 Result

B. Dynamic Programming Based Solution

Next I present a different solution by using the idea of Dynamic Programming [44]. Given a set of MSs $M=\{m_1,m_2,\ldots,m_n\}$ and a set of radio channels $H=\{h_1,h_2,\ldots,h_H\}$. For each MS-channel pair (m_i,h) , there exists a set of *feasible time intervals* $T_i^h=\{(i,h,t_j,t_k)\,|\,0\le t_j\le t_k\le\tau\}$ that can be used in a feasible schedule for m_i. Then, the set of feasible time intervals of all MS-channel pairs is $T = \{T_i^h : m_i \in M, h \in H\}$. With this information, our DP Scheduling Algorithm is listed as following:

Algorithm 11 DP Scheduling (M , H , T)

```
1: for each channel $h \in H$ do
2:     Construct a set $T^h$ of time interval element $(i, h, t_j, t_k)$ using channel $h$;
3:     $T^h \leftarrow \varnothing$; $A_{channel}[h] \leftarrow \varnothing$;
4:     for each MS $m_i \in M$ do
5:         $T^h \leftarrow T^h + T_i^h$;
6:     end for
7:     $A_{channel}[h] \leftarrow$ MITS ( $T^h$, $\tau$ );
8: end for
9: for each MS $m_i \in M$ do
10:    $T_i \leftarrow \varnothing$;
11:    for each channel $h \in H$ do
12:        $T_i \leftarrow T_i + T_i^h \cap A_{channel}[h]$;
13:    end for
14:    $A_{MS}[i] \leftarrow$ MITS ( $T_i$, $\tau$ );
15: end for
16: return $A = \{A_{MS}[i] \mid i \in \{1, 2, \ldots, n\}\}$.
```

We first fix the channel, and aim to find a maximum set of conflict-free feasible time slots for all MSs (Line 1 – Line 8). These time slots of using radio channel h are stored in $A_{channel}[h]$. Next step, we try to filter the found independent time slots, which are for all the MSs, for each MS (Line 9 - Line 15).

To find a maximum conflict-free time slot set, we propose a dynamic programming approach in Algorithm 12. Using dynamic programming scheme, Algorithm 12 can find a set of time-slots with maximum total time span while each time interval in the set does not conflict with others.

Algorithm 12 MITS (T , τ)

```
Construct an array $A[0:\tau]$, where each element $A[\tau_j]$ is a set of independent time
  interval element selected from $T$;
Construct an array $Value[\tau]$, where each element $Value[\tau_j]$ is the value which is the
  sum of timeslots of all time intervals in $A[\tau_j]$;
$A[0] \leftarrow \varnothing$; $Value[0] \leftarrow 0$;
for all $t = 1$ to $\tau$ do
    $Value[t] \leftarrow Value[t-1]$; $A[t] \leftarrow A[t-1]$;
    for each time interval element $(i, h, t_j, t_k) \in T$ do
        if $t = t_k$ then
            $w \leftarrow t_k - t_j$;
            if $Value[t] < Value[t-w] + w$ then
                $Value[t] \leftarrow Value[t-w] + w$; $A[t] \leftarrow A[t-w] \cup \{(i, h, t_j, t_k)\}$;
            end if
        end if
    end for
end for
return $A[\tau]$.
```

I use an example in Figure 44 to demonstrate how Algorithm 11 works. In this example, we have 2 MSs and 2 channels, and $\tau = 500$ s. For MS m_1 and m_2, the corresponding sets of time intervals are shown in Figure 44.

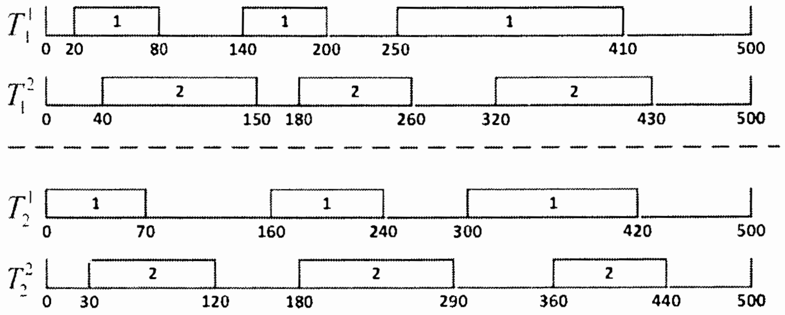


Figure 44 Example of time intervals

Then, using channel 1, we have

$$T^1 = \{(2,1,0,70),(1,1,20,80),(1,1,140,200),(2,1,160,240),(1,1,250,410),(2,1,300,420)\}$$

Similarly, using channel 2,

$$T^2 = \{(2,2,30,120),(1,2,40,150),(1,2,180,260),(2,2,180,290),(1,2,320,430),(2,2,360,440)\}$$

Then, in Line 7 of Algorithm 11, T^1 and T^2 are the inputs of Algorithm 12. For T^1, 2 arrays $A[\tau]$ and $Value[\tau]$ have been created in Algorithm 12, where $\tau = 500\,\text{s}$ in this case. First, $A[0]$ and $Value[0]$ have been initialized to $\varnothing$ and 0, respectively. For every t from 0 to 69, $A[t] = \varnothing$ and $Value[t] = 0$ since t is not end time of any time interval. When $t = 70$ (which is the end time of the first time interval), $Value[70]$ is initialized to $Value[69]$ firstly, then we can see that $Value[70] < Value[70-(70-0)]+(70-0)$, therefore, $Value[70]$ is changed to 70, and $\{(2,1,0,70)\}$ is added into $A[70]$ (Algorithm 12, Lines 6-13). After a sequence of these operations, we have $A[500] = \{(2,1,0,70),(2,1,160,240),(1,1,250,410)\}$. Therefore, the maximum set of times of using channel 1 is

$$A_{channel}[1] = \{(2,1,0,70),(2,1,160,240),(1,1,250,400)\}$$

Similarly, we can get

$$A_{channel}[2] = \{(1,2,40,150),(2,2,180,290),(1,2,320,430)\}$$

From Line 11 - 13, we obtain $T_1 = \{(1,2,40,150),(1,1,250,410),(1,2,320,430)\}$ and $T_2 = \{(2,1,0,70),(2,1,160,240),(2,2,180,290)\}$.

Next, we compute the schedule for each MS, $A_{MS}[1]$ and $A_{MS}[2]$, using input of T_1 and T_2, by Algorithm 12. Finally, we have the set of channel assignment schedules

$$A = \{\{(1,2,40,150),(1,1,250,410)\},\{(2,1,0,70),(2,2,180,290)\}\}$$

In Figure 45, the channel-time intervals assigned to the network are shown by the shadowed squares.

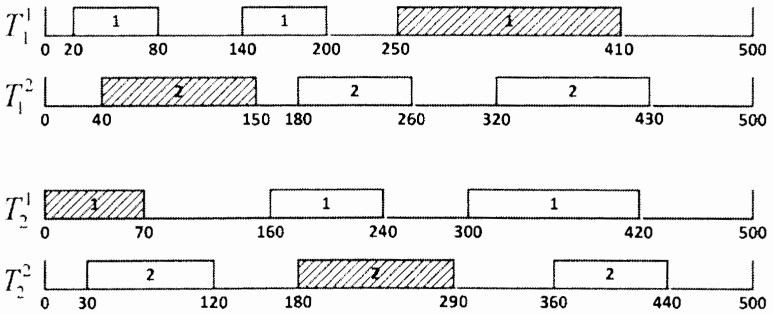


Figure 45 Example of channel assignment schedules

Numerical Results

In this section, I present numerical results to confirm the effectiveness of our solutions. I implemented the maximum independent set based solution (MSRIS Based) and our dynamic programming base scheme (DP Scheduling). I compare the results of MSRIS Based with those obtained from DP Scheduling. All our simulation runs were performed on a 2.8 GHz Linux PC with 1G bytes of memory. The metrics used for comparison are *success ratio* (the percentage of the MSs that are fully satisfied), *running time*, and *throughput* (represented by the total number of time slots allocated to the MSs). Our numerical results are showed in Figures 46-51. For each tested scenario, the result is the average from 10 runs.

First, I aim to test the effect of the number of MSs on the performances of our schemes. As Figure 46, Figure 47 and Figure 48 showed, Scenario 1, 2, and 3 were designed to compare the *success ratio*, *running time*, and *throughput*, respectively. $H = 20$, $\tau = 1000$. The number of MSs n was increased from 25 to 125. Scenario 1, in Figure 46, showed that DP Scheduling satisfied more MSs than MSRIS did. Also, as the number of MSs increased, the time intervals conflicting with others also increased. Consequently, the success ratio decreased. In Scenario 2, I tested the running time of our algorithms. For MSRIS, as number of MSs increased, the number of corresponding edges in the auxiliary graph also increased. Since the running time of MSRIS is mainly related to the number of edges in the graph, the running time also increased. The running time of DP Scheduling did not change much because it is mainly related to number of time intervals and the total time τ. As showed in Figure 47, DP Scheduling is faster than the MSRIS. In Figure 48, the throughput (total number of time slots) allocated by it to all the MSs is more than the throughput gained by MSRIS.

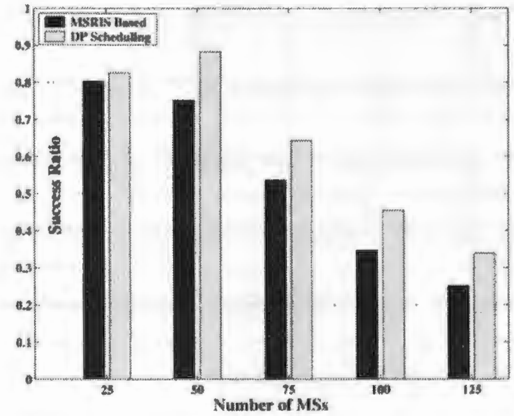


Figure 46 Success ratio (H = 20, τ = 1000)

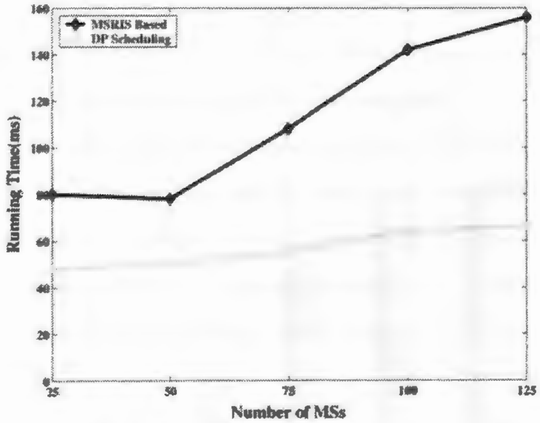


Figure 47 Running time (H = 20, τ = 1000)

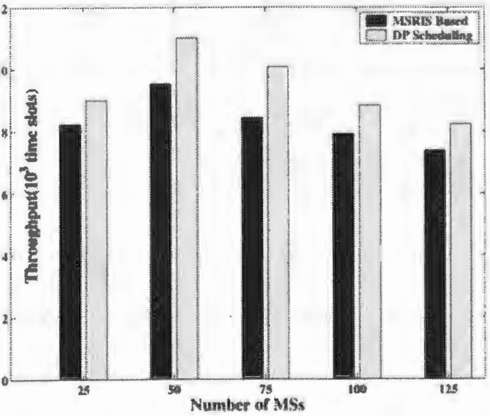


Figure 48 Throughput (H = 20, τ = 1000)

Next, in Figures 49-51, I target to test the impact of the number of channels on the performances of our schemes. In the tests, I used different number of channels which was increased from 10 to 50. Scenario 4, 5, and 6 are designed to test the success ratio, running time and the throughput respectively. The results showed that DP Scheduling generally has better performance than MSRIS. In Figure 49, it showed the success ratio increased for both schemes when the number of channels increased. The reason is that when the number of channels is increased, the conflicts between the time intervals could be decreased by using different channel. Consequently, the reduced number of conflicts leaded to more satisfied MSs. In Scenario 5, the efficiency of DP Scheduling is better than the one of MSRIS. As the number of channels increased, the running time of MSRIS decreased sharply.

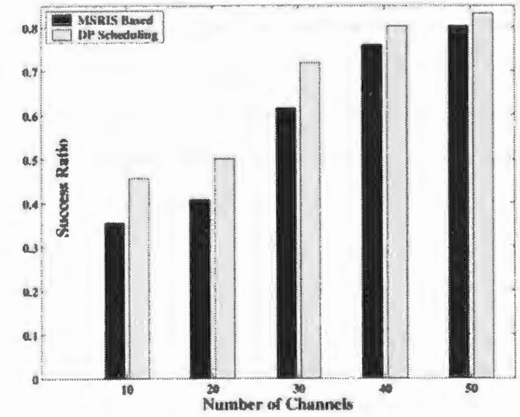


Figure 49 Success ratio (n = 75, τ = 1000)

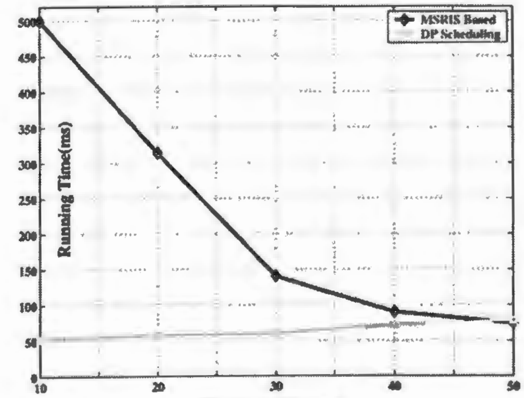


Figure 50 Running time (n = 75, τ = 1000)

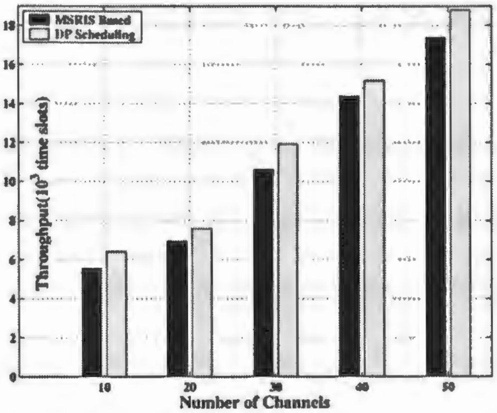


Figure 51 Throughput (n = 75, τ = 1000)

CHAPTER 5. CONCLUSION

In this thesis, I studied the Distance-Aware Relay Placement (DARP) problem, which seeks the multi-hop relay node placement with channel capacity constraint, in WiMAX mesh networks. I divided this problem into two sub-problems, Lower-tier Relay Coverage (LORC) problem and Minimum Upper-tier Steiner Tree (MUST) problem. For LORC problem, I proposed two approximation algorithms. For the MUST problem, I presented a minimum spanning tree based steinerization scheme, and proved this solution is an $8\frac{d_{max}}{d_{min}}$-approximation scheme. Then I presented an approximation framework of DARP by combining the solutions of the sub-problems. Numerical results confirmed our theoretical analysis.

I also studied the Multi-hop FAir Scheduling for Throughput Optimization (MFASTO) problem, which seeks the maximized minimum satisfaction ratio scheduling, in OFDMA-based multi-hop WiMAX mesh networks. For the MFASTO problem, I presented an Integer Linear Programming (ILP) formulation providing optimal solutions and a heuristic algorithm with better running time and scalability. Simulation results have been shown to justify the performance and efficiency of the solutions.

For the topic of scheduling for overwater communications, I studied how to provide effective overwater communication in wireless networks with cognitive radios. I defined the OVErwater Radio-Time Scheduling (OVERTS) problem. Two effective algorithms were proposed to solve the OVERTS problem. Simulation results have been shown to justify the performance and efficiency of the algorithms.

REFERENCES

[1] M. Andrews and L. Zhang, Scheduling algorithms for multi-carrier wireless data systems, *ACM Mobicom'2007*, September 2007.

[2] I. F. Akyildiz, W-Y Lee, M. C. Vuran and S. Mohanty, NeXt generation/dynamic spectrum access/cognitive radio wireless networks: a survey, *Computer Networks*, Vol. 50, No. 13, pp. 2127-2159, 2007.

[3] M. Alicherry, R. Bhatia and L. Li, Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks, *Proceeedings of ACM MobiCom'2005*, pp. 58-72.

[4] M. Awad and X. Shen, OFDMA based two-hop cooperative relay network resources allocation, *IEEE ICC*, 2008.

[5] N. Banerjee, M. Corner, D. Towsley, and B. Levine, Relays, base stations, and meshes: enhancing mobile networks with infrastructure, *ACM MobiCom'08*, pp. 81-91.

[6] G. Bernardi, P. Buneman, and M. Marina, Tegola tiered mesh network testbed in rural scotland, *Proceedings of ACM WINS-DR'2008*, pp. 9-16.

[7] V. Brik, E. Rozner, S. Banarjee and P. Bahl, DSAP: a protocol for coordinated spectrum access, *Proceedings of IEEE DySPAN'2005*, pp. 611-614.

[8] L. Cao and H. Zheng, Distributed spectrum allocation via local bargaining, *Proceedings of IEEE SECON'2005*, pp. 475-486.

[9] D. Chen, D. Du, X. Hu, G. Lin, L. Wang, and G. Xue, Approximations for steiner trees with minimum number of steiner points, *Journal of Global Optimization*, Vol. 18, pp. 17-33, 2000.

[10] F. Claude, R. Dorrigiv, S. Durocher, and R. Fraser, Practical discrete unit disk cover using an exact line-separable algorithm, *ISAAC'09*, pp. 45-54.

[11] T. Cover and A. E. Gamal, Capacity theorems for the relay channel, *IEEE Transactions on Information Theory*, Vol. 25, Issue 5, pp. 572-584, 1979.

[12] S. Deb, V. Mhatre, V. Ramaiyan, WiMAX relay networks: opportunistic scheduling to exploit multiuser diversity and frequency selectivity, *ACM MobiCom'2008*, pp. 163-174.

[13] P. Du, W. Jia, L. Huang, W. Lu, Centralized scheduling and channel assignment in multi-channel single-transceiver WiMAX mesh network, *IEEE WCNC'2007*, pp. 1734-1739.

[14] A. Efrat, S. P. Fekete, P. R. Gaddehosur, J. S. B. Mitchell, V. Polishchuk, and J. Suomela, Improved approximation algorithms for relay placement, *Lecture Notes In Computer Science*, Vol. 5193, pp. 356-367, 2008.

[15] R. J. Fowler, M. S. Paterson, and S. L. Tanimoto, Optimal packing and covering in the plane are NP-complete, *Information Processing Letters*, Vol. 12, pp. 133-137, 1981.

[16] M. R. Garey and D. S. Johnson, Computers and intractability: a guide to the theory of NP-completeness. *W. H. Freeman & Co.*, New York, 1979.

[17] V. Genc, S. Murphy, Y. Yu, and J. Murphy, IEEE 802.16J relay-based wireless access networks: an overview, *IEEE Wireless Communications*, Vol. 15, Issue. 5, pp. 56-63, 2008.

[18] X. Han, X. Cao, E. L. Lloyd, and C. C. Shen, Fault-tolerant relay node placement in heterogeneous wireless sensor networks, *IEEE INFOCOM'07*, pp. 1667-1675.

[19] Z. Han, Z. Ji, K. J. R. Liu, Non-cooperative resource competition game by virtual referee in multi-cell OFDMA networks, *IEEE Journal on Selected Areas in Communications*, Vol. 25, Issue 6, pp. 1079-1090, 2007.

[20] D. S. Hochbaum and W. Maass, Approximation schemes for covering and packing problems in image processing and VLSI, *Journal of the ACM*, Vol. 32, pp. 130-136, 1985.

[21] Y. T. Hou, Y. Shi, H. D. Sherali, and S. F. Midkiff, Prolonging sensor network lifetime with energy provisioning and relay node placement, *IEEE SECON'05*, pp. 295-304.

[22] L. Huang, M. Rong, L. Wang, Y. Xue, and E. Schulz, Resource allocation for OFDMA based relay enhanced cellular networks, *IEEE VTC'2007*, pp. 3160-3164.

[23] G. Li and H. Liu, On the capacity of the broadband relay networks, *Thirty-eighth Asilomar Conference on Signals, Systems and Computers*, Vol. 2, 2004, pp. 1318-1322.

[24] G. Li and H. Liu, Resource allocation for OFDMA relay networks with fairness constraints, *IEEE Journal on Selected Areas in Communications*, Vol. 24, No. 11, pp. 2061-2069, 2006.

[25] G. Li and H. Liu, Downlink radio resource allocation for multi-cell OFDMA system, *IEEE Transactions on Wireless Communications*, Vol. 5, Issue 12, pp. 3451-3459, 2006.

[26] B. Lin, P. Ho, L. Xie, and X. Shen, Optimal relay station placement in IEEE 802.16j networks, *ACM IWCMC'07*, pp. 25-30.

[27] B. Lin, P. Ho, L. Xie, and X. Shen, Relay station placement in IEEE802.16j dual-relay MMR networks, *IEEE ICC'08*, pp. 3437-3441.

[28] G. Lin and G. Xue, Steiner tree problem with minimum number of steiner points and bounded edge-length, *Information Processing Letters*, Vol. 69, pp. 53-57, 1999.

[29] E. Lloyd and G. Xue, Relay node placement in wireless sensor networks, *IEEE Transactions on Computers*, Vol. 56, pp. 134-138, 2007.

[30] H. Luo, R. Ramjeey, P. Sinhaz, L. Li, and S. Lu, UCAN: A unified cellular and AdHoc network architecture, *proceedings of ACM MobiCom'03.*

[31] N. Mustafa and S. Ray, PTAS for geometric hitting set problems via local search, *SCG'09*, pp. 17-22.

[32] S. Nahle, L. Iannone, B. Donnet, and N. Malouch, On the construction of WiMAX mesh tree, *IEEE Communications Letters*, Vol. 11, No. 12, 2007.

[33] O. Oyman, OFDMA^{2}A: A centralized resource allocation policy for cellular multi-hop networks, *IEEE Asilomar Conference on Signals, Systems and Computers*, November 2006.

[34] T. S. Rappaport, Wireless Communications: Principles and Practice, *Prentice Hall*, 2001.

[35] J. Pan, Y. T. Hou, L. Cai, Y. Shi, and S. X. Shen, Topology control for wireless mobile networks, *ACM MobiCom'03*, pp. 286-299.

[36] S. Rajagopalan and V. Vazirani, Primal-dual RNC approximation algorithms for set cover and covering integer programs, *SIAM Journal on Computing*, Vol. 28, pp. 525-540, 1998.

[37] A. Raniwala and T. Chiueh, Architecture and algorithms for an IEEE 802:11-based multi-channel wireless mesh network, *Proceedings of IEEE INFOCOM'2005*, pp. 2223-2234.

[38] A. Raniwala, K. Gopalan and T. Chiueh, Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks, *ACM Mobile Computing and Communications Review (MC2R)*, Vol. 8, No. 2, pp. 50-65, 2004.

[39] T. S. Rappaport, Wireless Communications: Principles and Practice (2nd Edition), *Prentice Hall*, 2002.

[40] Y. Shi and Y. T. Hou, Approximation algorithm for base station placement in wireless sensor networks, *IEEE SECON'07*, pp. 512-519.

[41] Y. Shi and Y. T. Hou, Theoretical results on base station movement problem for sensor network, *IEEE INFOCOM'08*, Phoenix, AZ, USA.

[42] A. Srinivas, E. Modiano, Joint node placement and assignment for throughput optimization in mobile backbone networks, *IEEE INFOCOM'08*, pp. 1130-1138.

[43] K. Sundaresan, W. Wang and S. Eidenbenz, Algorithmic aspects of communication in ad hoc networks with smart antennas, *Proceedings of ACM MobiHoc'2006*, pp. 299–309.

[44] K. J. Supowit. Finding a maximum planar subset of a set of nets in a channel, *IEEE Transaction on CAD of Integrated Circuits and Systems*, Vol. 6, No. 1. pp. 93-94, 1987.

[45] J. Tang, B. Hao, and A. Sen, Relay node placement in large scale wireless sensor networks, *Journal of Computer Communications*, Vol. 29, No. 4, pp. 490-501, 2006.

[46] J. Tang, S. Misra and G. Xue, Joint spectrum allocation and scheduling for fair spectrum sharing in cognitive radio wireless networks, *Computer Networks*, Vol. 52, No. 11, pp. 2148-2158, 2008.

[47] J. Tang, G. Xue and W. Zhang, Interference-aware topology control and QoS routing in multi-channel wireless mesh networks, *Proceedings of ACM MobiHoc'2005*, pp. 68-77.

[48] Z. Tao, K. H.Teo, and J. Zhang, Aggregation and concatenation in IEEE 802. 16j mobile multihop relay (MMR) networks, *Mobile WiMAX Symposium*, pp. 85-90, 2007.

[49] K. Xu, H. Hassanein, G. Takahara, and Q. Wang, Relay node deployment strategies in heterogeneous wireless mobile networks: multiple-hop communication case, *IEEE SECON'05*, pp. 575-585.

[50] Y. Xu, S. Wan, J. Tang and R. S. Wolff, Interference aware routing and scheduling in wireless backhaul networks with smart antennas, *IEEE SECON'2009.*

[51] H. Viswanathan, S. Mukherjee, Performance of cellular networks with relays and centralized scheduling, *IEEE Transactions on Wireless Communications*, Vol. 4, Issue 5, pp. 2318-2328, 2005.

[52] Y. Yu, S. Murphy, and L. Murphy, A clustering approach to planning base station and relay station locations in IEEE 802.16j multi-hop relay networks Communications, *Proceedings of ICC'08*, pp. 2586-2591.

[53] Y. Yuan, P. Bahl, R. Chandra, T. Moscibroda and Y. Wu, Allocating dynamic time-spectrum blocks in cognitive radio networks, *Proceedings of ACM MobiHoc'2007*, pp. 130-139.

[54] W. Zhang, F. Kandah, J. Tang and K. Nygard, Interference-aware robust topology design in multi-channel wireless mesh networks, *IEEE CCNC'2010*, pp. 1-5.

[55] W. Zhang, G. Xue, and S. Misra, Fault-tolerant relay node placement in wireless sensor networks: problems and algorithms, *Proceedings of IEEE INFOCOM'07*, pp. 1649-1657.

[56] Q. Zhao, L. Tong and A. Swami, Decentralized cognitive MAC for dynamic spectrum access, *Proceedings of IEEE DySPAN'2005*, pp. 224-232.

[57] IEEE 802.16 Working Group, Part 16: Air interface for fixed broadband wireless access systems, *IEEE Standard*, 2004.

[58] 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 Standrd*, 2005.

[59] IEEE 802.16 Working Group, Part 16: Air interface for fixed and mobile broadband wireless access systems–multihop relay specification, *IEEE Draft Standard*, 2007.

[60] IEEE 802.16j-06/014r1, Harmonized definitions and terminology for 802.16j Mobile Multihop Relay, *http://www.ieee802.org/16/relay/index.html*, October 2006.

[61] Gurobi Optimization, *http://www.gurobi.com/*.

[62] Space and Naval Warfare Systems Center, Advanced Refractive Effects Prediction System (AREPS) (Version 3.6), 2006, http://areps.spawar.navy.mil/