In: Journal of Communications Research
Volume 2, Issue 2-3, pp. 1-20
ISSN: 1935-3537

Scheduling Problems and Solutions in WiMAX Mesh Networks

Jia-Ming Liang*[1], *Jen-Jee Chen*[2], *Yu-Chee Tseng*[1], *and Bao-Shuh Paul Lin*[1][*]
[1] Department of Computer Science
National Chiao-Tung University, Hsin-Chu, 30010, Taiwan
[2] Department of Electrical Engineering
National University of Tainan, Tainan 70005, Taiwan

Abstract

WiMAX is developed to support large-scale wireless broadband access. The IEEE 802.16d defines two types of mesh scheduling mechanisms: One is the centralized scheduling mechanism; the other is the distributed scheduling mechanism. This chapter focuses on the centralized scheduling, which aims to schedule the traffic to and from the *base station (BS)*. In more detail, for centralized scheduling, the BS coordinates the resource allocation of all *subscribe stations (SSs)* in a mesh network. Initially, each SS sends the bandwidth demand to the BS. Then, the BS is responsible to schedule and grant resources for them. Since the BS has the global information and thus can optimize the scheduling assignments. In this chapter, we discuss the scheduling problems and their solutions, which cover the issues of how to improve network throughput, how to guarantee the fairness, and how to exploit multi-channel properties. The comparison of these scheduling solutions is also given in the end of the chapter.

Keywords: centralized scheduling, IEEE 802.16d, mesh network, minislot allocation, WiMAX.

1. Introduction

WiMAX is an emerging wide-range wireless access technology for solving the last-mile communication problem, bridging the Internet and wireless local-area networks, and supporting broadband multimedia communication services [3, 8]. To support a huge area such as metropolis or large islands, IEEE 802.16d [5] provides the mesh mode to inherit the *point-to-multipoint (PMP)* networks. Under the mesh mode, all *subscribe stations (SSs)* are organized in an ad hoc fashion and the traffic of each SS can reach the gateway BS through a multihop manner. In the standard, two types of scheduling mechanisms are defined: 1) Centralized scheduling and 2) Distributed scheduling. The centralized scheduling focuses on the Internet traffic in and out of the network through the BS. The distributed scheduling focuses on the intranet traffic. Since most of the studies focus on the centralized scheduling and it conducts better performance than the distributed scheduling, this chapter will focus

[*]E-mail address: {jmliang, yctseng}@cs.nctu.edu.tw, jjchen@mail.nutn.edu.tw, bplin@mail.nctu.edu.tw

on the centralized scheduling. Since the WiMAX mesh network supports multi-hop transmissions, which inherits more characteristic features and constraints compared to the PMP mode and thus makes the scheduling problem more challenging and interesting. This chapter provides a comprehensive survey of the scheduling problems and solutions in WiMAX mesh networks, which covers the following research issues:

- **Network throughput:** Since the objective of WiMAX mesh networks is to provide broadband network access, we will introduce several scheduling schemes that target at improving network throughput. The concepts of control overhead reduction and concurrent transmissions are adopted to help enhance throughput.

- **Fairness:** Since the WiMAX mesh network is organized in a multi-hop manner, to provide the fairness among SSs with different hop counts to the BS, we will introduce several scheduling schemes that aim at maintaining fairness among SSs. The concepts of weight assignment and token allocation will be conducted to help improve the fairness.

- **Channel Assignment:** Since the WiMAX mesh network supports multiple channels to reduce the secondary interference, we will introduce several scheduling schemes that focus on channel assignments. The concepts of minimal link coloring will be exploited to help reduce the number of assigned channels.

The rest of this chapter is organized as follows: Some background knowledge of WiMAX mesh networks is given in Section 2.. Section 3. presents the centralized scheduling solutions for WiMAX mesh networks under the issues of network throughput, fairness maintenance, and multi-channel assignments.

2. WiMAX Mesh Networks

Below, we give an overview of WiMAX mesh networks, which covers the topics of network architecture, access technology in the physical layer, frame structure, and QoS service classes.

2.1. Network Architecture

A mesh network consists of one BS and multiple *static* SSs, as shown in Fig. 1. Specified by the IEEE 802.16d standard, all SSs will be organized in an ad hoc manner to cover a huge area. Two SSs can communicate with each other if they are within each other's transmission range. Each SS can act as either an endpoint or a router to relay data for its neighbors. For centralized scheduling schemes, the BS is responsible for managing the radio resource. All SSs have to send their requests containing traffic demands and link qualities to the BS. Then, with the topology information and SSs' requests, the BS will construct a routing tree for SSs to transmit/receive their data, as shown in Fig. 1. It can be observed that more than one transmissions could coexist at the same time if any two of them are far enough away from each other.

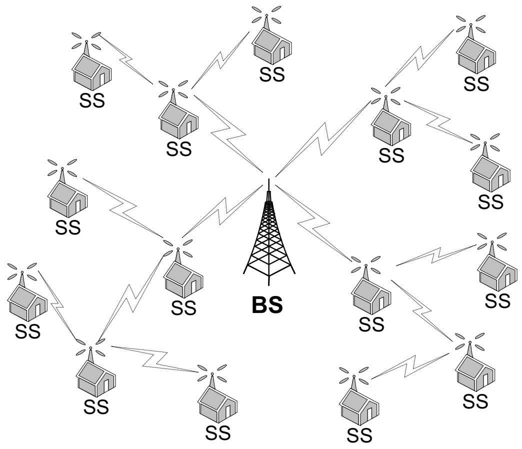


Figure 1. The network architecture of the WiMAX mesh network. Under the mesh architecture, the BS constructs a routing tree for SSs to transmit/receive their data.

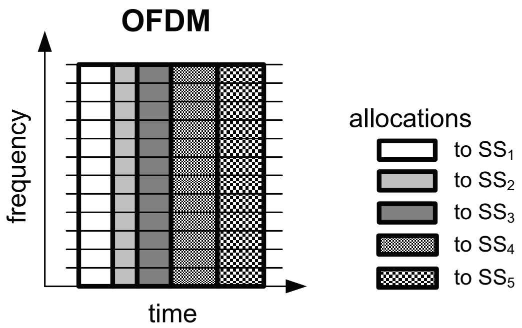


Figure 2. The OFDM access technology is adopted in the WiMAX mesh physical layer, where each SS has the full control of all subcarriers at different times.

2.2. Access Technology

The WiMAX mesh network adopts *orthogonal frequency division multiplexing (OFDM)* as the access technology in the physical layer, as shown in Fig. 2. OFDM supports *non-line of sight (NLOS)* communications and multicarrier transmissions, where each SS is given the complete control of all subcarriers. The BS adopts the concept of *time division multiple access (TDMA)* to share the radio resource among all SSs. In other words, for multiple SSs that are within each other's transmission range, only one SS is allowed to access the channel at any time. Therefore, the scheduler only needs to determine which time slot should be allocated to which SS.

2.3. Frame Structure

Taking OFDM as the access technology in the physical layer, the frame resource under the mesh network is modeled as an one-dimensional array over the time domain. In each frame, there are one *control subframe* and one *data subframe*, where the control subframe

Table 1. The control messages and scheduling parameters used for centralized scheduling.

message/parameters	description
MSH-NENT	network entry message
MSH-NCFG	network configuration message
MSH-CSCH	centralized scheduling message (for requesting and granting)
MSH-CSCF	centralized scheduling configuration message
MSH-CTRL-LEN	the length of the total control subframe

Table 2. The six MCSs supported by WiMAX: Using different MCS levels, each slot can carry different amount of data and each MCS requires a minimum *signal to interference plus noise ratio (SINR)*.

level	MCS	data carried by each slot	minimum SINR
1	QPSK 1/2	48 β	6 dBm
2	QPSK 3/4	72 β	8.5 dBm
3	16QAM 1/2	96 β	11.5 dBm
4	16QAM 3/4	144 β	15 dBm
5	64QAM 2/3	192 β	19 dBm
6	64QAM 3/4	216 β	21 dBm

† β is a constant in bits.

is used for broadcasting the network information and the data subframe is used for data transmission or reception. Specifically, two types of the control subframes are defined, as shown in Fig. 3. A type-1 control subframe (also called the network control subframe) consists of the network formation information, such as the mesh network entry message (MSH-NENT) and the mesh network configuration message (MSH-NCFG). A type-2 control subframe (also called the *schedule control subframe*) is used to specify the resource allocation in the following data subframe. In addition, the schedule control subframe is further divided into two blocks. The first is used for exchanging centralized scheduling messages (MSH-CSCH). The second is used for exchanging the distributed scheduling messages (MSH-DSCH), where its length is defined by the number of DSCH opportunities in schedule control subframe (MSH-DSCH-NUM) which is contained in the MSH-NCFG message. The control subframe is sent using QPSK-1/2 (quaternary phase shift keying) modulation and the total length of the control subframe is defined by the control subframe length (MSH-CTRL-LEN) which is contained in the MSH-NCFG message. On the other hand, for the data subframe, it is partitioned into centralized and distributed data subframes. The basic unit in a data subframe is called a *minislot*, which contains a specific number of OFDM symbols and is determined by

$$\frac{\text{OFDM Symbols per frame} - 7 \times \text{MSH-CTRL-LEN}}{256}. \tag{1}$$

Table 1 summarizes the control messages and scheduling parameters used for centralized scheduling in the WiMAX mesh mode. When scheduling, the resource allocation unit for each SS is a *burst*, which contains one or multiple continuous minislots. The allocated

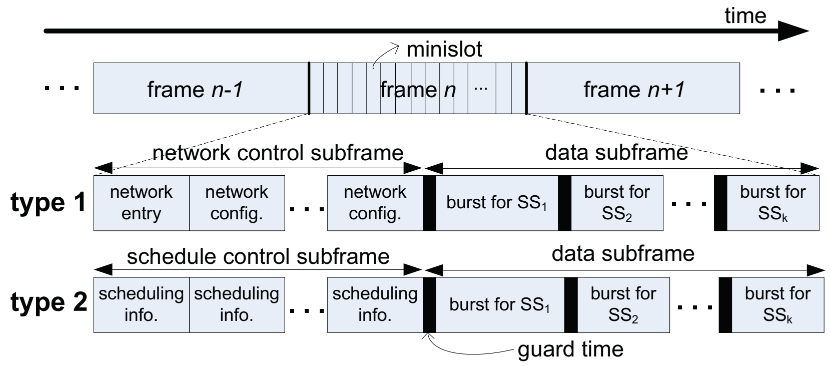


Figure 3. The frame structures of the WiMAX mesh network, which is modeled by an one-dimensional array over the time domain.

bursts are specified in MSH-CSCH messages, which contains the access information, such as the available *modulation and coding scheduling (MCS)*, as shown in Table 2, and indicates that the bursts are allocated for transmission or reception, to which SS, and on which minislots. For each burst, it requires a *guard time* in front of it to conduct time synchronization and avoid the previous transmission interfering with the following transmission because of propagation delay. Such a guard time is usually viewed as transmission overhead because it does not carry any SS's data. It can be observed that the transmission overhead caused by guard times will degrade network performance and thus how to alleviate this problem is a critical issue.

2.4. QoS Service Classes

To satisfy the different requirements of various data traffics, WiMAX mesh networks support five types of QoS service classes:

Unsolicited grant service (UGS): The UGS class provides fixed periodic bandwidth allocation for *constant bit rate (CBR)* traffic such as E1/T1 circuit emulation. Each SS only needs to negotiate with the BS about the QoS parameters such as the maximum sustained rate, maximum latency, and tolerated jitter at the first time when the connection is established. Then, no further negotiation is required. The UGS class can guarantee the maximum latency for those delay-critical real-time services. However, the radio resource would be wasted if the granted traffic does not fully utilize the allocated bandwidth.

Real-time polling service (rtPS): The rtPS class supports *variable bit rate (VBR)* traffic such as compressed videos. Unlike UGS, the BS has to periodically poll each SS for its QoS parameters such as the maximum sustained rate, maximum latency, tolerated jitter, and minimum reserved rate. The benefit is that the BS can adjust bandwidth allocation according to the real demands of traffics. However, periodical polling spends the radio resource.

Extended real-time polling service (ertPS): The ertPS class is specially designed for *voice over IP (VoIP)* with silence suppression, where no traffic is sent during silent periods. Both ertPS and UGS share the same QoS parameters. The BS will allocate the bandwidth equal to the maximum sustained rate when the VoIP traffic is active and reserve only the polling bandwidth when it becomes silent. In this way, the BS only has to poll SSs during the silent period to determine whether their VoIP traffics become active again.

Non-real-time polling service (nrtPS): The nrtPS class considers the non-real-time traffic with minimum reserved rate. The *file transfer protocol (FTP)* is one representative example. The BS will preserve bandwidth according to the minimum reserved rate to avoid starving the non-real-time traffic.

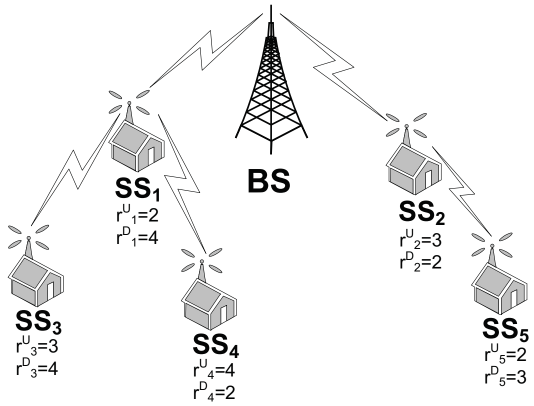
Best effort service (BE): All other types of traffic belong to this service class. The BS will distribute the remaining bandwidth (after allocating to the traffic of the previous four service classes) to the traffic of the BE class, so there is no guarantee of throughput or delay.

3. Scheduling Solutions in WiMAX Mesh Networks

In WiMAX mesh networks, the BS and all SSs form a routing tree rooted at the BS for communications. The BS will determine the resource allocation of all SSs in the network and propagate the transmission schedules to the network. By periodically broadcasting MSH-CSCF message, the BS can maintain the routing tree of the network while all SSs can have the information about the routing tree. Once a new SS adds, the routing tree is updated and the BS will inform the network by broadcasting a new MSH-CSCF message accordingly.

According to the standard, the BS determines the resource allocation for SSs by the following procedure which includes three steps. In the first step, SSs can transmit MSH-CSCH:Request messages to request the resource. The transmission order is from leaves to the root. An SS will combine the request from its children to its own MSH-CSCH message, and then transmit the message to its parents. Thus, all the request messages in the network are collected by the BS in a bottom up manner. Each request message contains the corresponding SS's SS ID and its acceptable transmission and reception rates. In the second step, the BS performs a scheduling algorithm to determine the transmission schedules. In the third step, the BS broadcasts an MSH-CSCH:Grant message containing the transmission schedules to all SSs. Upon the receipt of the MSH-CSCH:Grant message, each SS executes a common algorithm to derive the uplink and downlink minislots allocated to each of them. Thus, each SS can deliver their data with these minislots of resource.

Note that the standard defines only a framework for the scheduling of the WiMAX mesh network and the scheduling algorithm is left undefined. But it gives an example to show how the resource may be scheduled and allocated to SSs. In the following, we show the example given in the standard. Consider five SSs in the network and the routing tree is shown in Fig. 4. These SSs have both the uplink and downlink bandwidth demands of $\{2,3,3,4,2\}$ and $\{4,2,4,2,3\}$ in Mbps, respectively, to request their parents. Through a bottom up manner, the BS gathers all the bandwidth demands of the SSs. Then, the BS determines the resource of each SS being proportional to the amount of bandwidth demands gathered by itself. For example in uplink transmission, SS_1 has an aggregated bandwidth demand $\widehat{R}_1^U = 9$ (2 from itself, 3 from SS_3, and 4 from SS_4), and SS_2, SS_3, SS_4, and SS_5 have aggregated bandwidth demands $\widehat{R}_2^U = 5$, $\widehat{R}_3^U = 3$, $\widehat{R}_4^U = 4$, and $\widehat{R}_5^U = 2$, respectively. On the other hand, in downlink transmission, $SS_1 \sim SS_5$ have downlink bandwidth demands $\widehat{R}_1^D = 10$, $\widehat{R}_2^D = 5$, $\widehat{R}_3^D = 4$, $\widehat{R}_4^D = 2$, and $\widehat{R}_5^D = 3$, respectively. To sum up all these



SS ID	uplink grant fraction	downlink grant fraction
1	9/47	10/47
2	5/47	5/47
3	3/47	4/47
4	4/47	2/47
5	2/47	3/47

Figure 4. An example of how the BS may grant the bandwidth to each SS. r_i^U and r_i^D are the bandwidth demands of SS_i in the uplink and downlink directions, respectively. This case shows that the BS grants each SS the bandwidth being proportional to its aggregated requests for both uplink and downlink transmissions.

demands, we will get $\sum_{i=1..5}\widehat{R}_i^U + \sum_{i=1..5}\widehat{R}_i^D = 47$. Therefore, $SS_1 \sim SS_5$ are assigned resource $\frac{9}{47}B$, $\frac{5}{47}B$, $\frac{3}{47}B$, $\frac{4}{47}B$, and $\frac{2}{47}B$ in the uplink direction and $\frac{10}{47}B$, $\frac{5}{47}B$, $\frac{4}{47}B$, $\frac{2}{47}B$, and $\frac{3}{47}B$ in the downlink direction, respectively, where B is the total bandwidth.

To summarize, based on the information of network topology and bandwidth demand of each SS, the BS can determine the transmission schedules for SSs. We can see that the scheduling solution shown in the example is quite trivial and neglects several key features of the WiMAX mesh network, such as the spatial reuse, transmission overhead reduction, and channel assignments. We will discuss these features in the following sections.

3.1. Weight-Based Scheduling

To improve transmission efficiency, the work of [7] proposes to use spatial reuse while consider the fairness of resource allocation. Two phases are involved in the work. The first phase is to determine the *allocation order* of SSs to maintain the fairness of SSs. This phase is executed in the BS side. The second phase is to conduct the scheduling assignments based on the bandwidth demands of SSs and the allocation order given by the BS. This phase is executed by the common algorithm of SSs. In this phase, the network topology is known to the SSs such that they can exploit spatial reuse to improve network throughput.

To determine the allocation order, this work defines the *satisfaction index* for each SS to calculate each SS's bandwidth satisfaction degree. The satisfaction index s_i of $\mathrm{SS}_i, i = 1..n$, is defined as the average allocated bandwidth over a period of time T divided by a given weight W_i, which can be represented as follows.

$$s_i(x) \triangleq \frac{(\sum_{y=x-T}^{x-1} B_i(y))/T}{W_i}, \tag{2}$$

where x is the current frame index, $B_i(y)$ is the allocated bandwidth of SS_i in frame y, T is a satisfaction window, and W_i is the weight of SS_i. Note that the W_i is adjustable which can

Table 3. The schedule matrix.

minislot	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
SS ID	SS_1 SS_5	SS_1 SS_5	SS_1	SS_1	SS_1	SS_1	SS_1	SS_1	SS_1	SS_3 SS_2	SS_3 SS_2	SS_3 SS_2	SS_4 SS_2	SS_4 SS_2	SS_4	SS_4

reflect the priority of an SS according to the QoS requirement. The weight W_i of each SS_i is its weight w_i plus the sum of all weights of its children, which is defined as follows.

$$W_i = w_i + \sum_{j \in i\text{'s children}} W_j. \quad (3)$$

Thus, the BS determines the allocation order by sorting SSs' satisfaction index in increasing order and broadcasts the results by using MSH-CSCF messages. On receiving the information, each SS calculates its minislots for transmissions by the phase 2 of the scheme.

When calculating the minislots, SSs will conduct the whole transmission schedules of the network. This work assumes that SSs have the information of the network topology, each SS calculates the allocated minislots by maintaining a *schedule matrix* and a *collision matrix*. The schedule matrix is an array which records who will transmit in each minislot. The collision matrix is an array recording which SS will be interfered in the corresponding minislots. Then, SSs are allocated minislots according to the bandwidth demands and following the allocation order. Each SS is allocated minislots from the earliest minislot where it does not appear in the collision matrix until the demand is satisfied. Note that during the resource allocation, each SS is not allowed to use the minislots where it appears in the collision matrix to avoid interference. Then, the SS will be added into the schedule matrix according to its minislot allocation and all the SSs who cause interference to this SS will be added to the collision matrix accordingly. This operation will be repeated until all SSs are allocated. Finally, the schedule matrix is the actual transmission schedules.

Below, we show an example of the scheme. Consider the uplink bandwidth demands of the five SSs in Fig. 4. Assume that we have the allocation list of $\{SS_1, SS_3, SS_4, SS_2, SS_5\}$. Initially, both the schedule and collision matrices are empty. First, SS_1 will be allocated 9 minislots and added to the schedule matrix at minislot $1 \sim 9$. Then, all the SSs who interfere SS_1, such as $\{SS_2, SS_3, SS_4\}$, are added to the collision matrix at minislot $1 \sim 9$. Next, SS_3 will be allocated 3 minislots and added to the schedule matrix at minislot $10 \sim 12$ since it appears in the collision matrix at minislot $1 \sim 9$. Then, SS_1 and SS_4, who cause interference to SS_3, will be added in the collision matrix accordingly. Consequently, SS_4 will be allocated 4 minislots and added to the schedule matrix at minislot $13 \sim 16$ since it appears in the collision matrix at minislot $1 \sim 12$. Now, when allocating SS_2, since it appears in the collision matrix at minislot $1 \sim 9$, SS_2 will be scheduled at minislot $10 \sim 14$ and SS_5 will be added to the collision matrix accordingly. For SS_5, it will be allocated at minislot $1 \sim 2$. The final schedule and collision matrices are shown in Table 3 and Table 4, respectively.

In summary, this scheme can realize fairness among SSs while improve the network throughput by exploiting spatial reuse. However, it may incur more transmission delay because the parent SSs may be allocated resource earlier than their children. For example, above scheduling results show that SS_1 (the parent) is allocated earlier than SS_3 and SS_5

Table 4. The collision matrix.

minislot	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
SS ID	SS_2 SS_3 SS_4	SS_2 SS_3 SS_4	SS_2 SS_3 SS_4	SS_2 SS_3 SS_4	SS_2 SS_3 SS_4	SS_2 SS_3 SS_4	SS_2 SS_3 SS_4	SS_2 SS_3 SS_4	SS_2 SS_3 SS_4	SS_1 SS_4 SS_5	SS_1 SS_4 SS_5	SS_1 SS_4 SS_5	SS_1 SS_3 SS_5	SS_1 SS_3 SS_5	SS_1 SS_3	SS_1 SS_3

(the children). In this case, these children's data will be relayed by their parents in the next round. We will discuss how to address this problem in the next section.

3.2. Traffic-Aware Link Scheduling

The work of [11] proposes a scheduling scheme to exploit the spatial reuse and consider the variation of traffic loads in each minislot, which can capture the traffic and thus can maintain the exact transmission order. Given a routing tree, the scheduling scheme determines the active links in each minislot t based on the traffic loads of SSs in the minislot. Here, the traffic load of each link is the traffic load of the transmitting SS in the routing tree. In each minislot t, all links are put in a set, named $AvailableLink$, initially. Then, it iteratively activates the link with the maximal traffic load from $AvailableLink$ and then removes the links which interfere to the selected link from $AvailableLink$. This operation is repeated until $AvailableLink$ is empty. Thus, the scheme is easy to implement and well improve the spectral efficiency. We use the uplink bandwidth demands of the five SSs in Fig. 4 as an example for the scheme. Initially, all links are in $AvailableLink$ due to their positive traffic loads. In minislot 1, link (SS_4, SS_1) will be activated since SS_4 has the maximal traffic load (that is, 4). Then, the links (SS_3, SS_1), (SS_1, BS) will be removed from $AvailableLink$ due to interfering to SS_4 and SS_1. Next, link (SS_2, BS) will be activated because it becomes the one with the maximum traffic load (that is, 3). Then, link (SS_5, SS_2) will be removed from $AvailableLink$ due to interference. At this point, $AvailableLink$ is empty and thus two links (SS_4, SS_1) and (SS_2, BS) are scheduled at minislot 1. After that, the traffic loads of SS_4 and SS_2 are updated by decreasing 1 and the traffic loads of the corresponding receivers SS_1 and the BS are updated by increasing 1 accordingly. In minislot 2, since more than one link has the maximum traffic load (that is, 3), the link with the smaller index is choose, such as the link (SS_1, BS), to be activated. Then, links (SS_3, SS_1), (SS_4, SS_1), and (SS_2, BS) will be removed from $AvailableLink$ due to interference. Next, link (SS_5, SS_2) will be activated because it becomes the one with the maximum traffic load (that is, 3). Now, $AvailableLink$ is empty and thus two links (SS_1, BS) and (SS_5, SS_2) are scheduled at minislot 2. Table 5 shows the variation of traffic loads of SSs in each minislot. The final transmission schedules is shown in Table 6.

To summarize, the scheduling scheme can well assign the transmission schedules for SSs to fit their traffic loads and solve the problem of transmission delay. However, this scheme has to schedule links minislot by minislot, which is time inefficient. In the next section, a batch-based allocation scheme will be introduced. In addition, more kinds of link selection criteria will be studied.

Table 5. The variation of traffic loads of SSs in each minislot by the traffic-aware link scheduling scheme. Here, we use load_i as the traffic load of SS_i and use $(\cdot)$ to show the active transmission at a minislot. In this case, SS_2 and SS_4 are scheduled at minislot 1 and their traffic loads are updated by decreasing 1, i.e., $(3) \rightarrow 2$ for SS_2 and $(4) \rightarrow 3$ for SS_4. In minislot 2, SS_1 and SS_5 are scheduled and thus their traffic loads are updated by 2 and 1 accordingly. This scheduling is terminated at minislot 17 and the scheduling length is 16.

minislot	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
load_1	2	(3)	2	(3)	2	(3)	(2)	1	(2)	1	(2)	1	(2)	(1)	0	(1)	0
load_2	(3)	2	(3)	2	(3)	2	2	(2)	1	(1)	0	0	0	0	0	0	0
load_3	3	3	(3)	2	2	2	2	(2)	1	1	1	(1)	0	0	0	0	0
load_4	(4)	3	3	3	(3)	2	2	2	2	(2)	1	1	1	1	(1)	0	0
load_5	2	(2)	1	(1)	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6. The final schedules of the traffic-aware link scheduling scheme.

minislot	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
SS ID	SS_2 SS_4	SS_1 SS_5	SS_2 SS_3	SS_1 SS_5	SS_2 SS_4	SS_1	SS_1	SS_2 SS_3	SS_1	SS_2 SS_4	SS_1	SS_3	SS_1	SS_1	SS_4	SS_1

3.3. Token-Based link Scheduling

The work of [4] proposes a token-based link scheduling scheme to maximize the spatial reuse and tries to further reduce the scheduling cycle. Initially, the scheme assigns each SS a *service token* with the size proportional to its traffic demand. Only the link with non-zero token can be scheduled which can prevent starvation among SSs. Then, a link selection algorithm is performed to determine the active links in each minislot based on the service tokens. Specifically, given n SSs and their traffic demands $r_i^U, i = 1..n$, in a scheduling tree (here we consider only the uplink transmission for simplicity). The scheduler assigns each $\text{token}_i = r_i^U / G$ to SS_i, where G is the *greatest common divisor (GCD)* of $r_i^U, i = 1..n$. By dividing their traffic demands to the GCD, this scheme can further reduce the scheduling cycle and allocate resource to SSs in a time efficient way. We show an example below. Assume that there are five SSs with traffic demands of 4, 6, 6, 8, and 4 Mbps. The scheme will assign token values of 2, 3, 3, 4, and 2 to the five SSs, respectively, due to $G = 2$. Thus, the scheduling cycle is reduced to half, compared with the service token assignments of 4, 6, 6, 8, and 4. Initially, the scheduling scheme starts from an empty schedule matrix. Then, the scheme iteratively chooses the active links by the link selection criteria (we will describe later on) in current minislot. Then, the service tokens of the transmitters and receivers of these links are updated by decreasing one and increasing one, respectively. These operations are repeated until all links are with zero service token.

Four criteria for the link selection are further investigated in this work. They are 1) random, 2) minimal interference, 3) nearest to the BS, and 4) farthest to the BS. The random selection schedules the links randomly. The minimal interference selection chooses the links with the minimal number of interfered SSs. The selections of the nearest and farthest

Table 7. The variation of service tokens for SSs in each minislot by the link selection criterion: Nearest to the BS. In minislot 1, SS_1 and SS_5 are scheduled in turn because they are the nearest ones to the BS. Then, their service tokens are updated by 1 accordingly. The transmission schedules at minislot 2 are the same as that at minislot 1. In minislot 3, since SS_2 and SS_3 become the nearest ones in turn to the BS and are with non-zero tokens, SS_2 and SS_3 are scheduled. These operations are terminated at minislot 17.

minislot	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
$token_1$	(2)	(1)	0	(1)	0	(1)	0	(1)	0	(1)	0	(1)	0	(1)	0	(1)	0
$token_2$	3	4	(5)	4	(4)	3	(3)	2	(2)	1	(1)	0	0	0	0	0	0
$token_3$	3	3	(3)	2	(2)	1	(1)	0	0	0	0	0	0	0	0	0	0
$token_4$	4	4	4	4	4	4	4	4	(4)	3	(3)	2	(2)	1	(1)	0	0
$token_5$	(2)	(1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8. The final transmission schedules by the link selection criterion: Nearest to the BS.

minislot	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
SS ID	SS_1 SS_5	SS_1 SS_5	SS_2 SS_3	SS_1	SS_2 SS_3	SS_1	SS_2 SS_3	SS_1	SS_2 SS_4	SS_1	SS_2 SS_4	SS_1	SS_4	SS_1	SS_4	SS_1

to the BS choose the links with the minimal and maximal hop counts to the BS. When more than one SS have the same property, the SS with the smaller ID is chosen to break the tie. Below, we show the link scheduling of the nearest to the BS for example. We still use the uplink bandwidth demands of the five SSs in Fig. 4 as an example. In minislot 1, SS_1 is scheduled since SS_1 and SS_2 are the nearest to the BS (that is, 1 hop), but SS_1 has the smaller ID than SS_2. Then, SS_2, SS_3, and SS_4 are interfered by SS_1, so the next one to be scheduled is SS_5. Thus, the service tokens of SS_1 and SS_5 are decreased by 1 and the service tokens of the BS and SS_2, which are the receivers, are increased by 1. Similarly, in minislot 2, SS_1 and SS_5 are scheduled. Now, in minislot 3, the scheme chooses SS_2 to schedule because SS_2 is the nearest to the BS and is with non-zero token. Consequently, SS_3 and SS_4 are the next nearest to the BS and are interference-free to SS_2. Compared to SS_4, SS_3 has a smaller ID than SS_4 and thus SS_3 is scheduled. Therefore, SS_2 and SS_3 are scheduled in minislot 3. The remaining variation of SSs' service tokens and the final transmission schedules are shown in Table 7 and Table 8, respectively.

By using service tokens and four kinds of link selection criteria, the schedule cycle and link scheduling can be further investigated. In the following, we will introduce the work considering the transmission overhead and the maximal concurrent transmissions.

3.4. Maximal Concurrency Scheduling

By adopting the nature of spatial reuse under the mesh networks, the work of [12] proposes two strategies to allow more concurrent transmissions to reduce the scheduling length. Two transmissions are allowed to coexist if they do not interfere with each other. The first strategy is to try to find out the concurrent transmissions that can transmit the maximum amount of data. When all SSs have the same transmission rate, this strategy will find the maximum number of concurrent transmissions in each iteration. Then, the BS selects the minimum burst length, l_{min}, among these transmissions and allocates a burst for each transmission with the length of l_{min} and reserves a transmission overhead in front of each burst. The above operations are repeated until all data in each SS's queue are consumed.

Let q_i, c_i, and h_i be the queue length, transmission rate, and hop count from the BS of an SS_i, respectively. The second strategy proposes to adopt the following six criteria to select concurrent transmissions (note that in the following six criteria, we use $\widehat{S}$ to describe a set of SSs which are able to do concurrent transmissions):

1. The total queue length of the SSs in $\widehat{S}$:
   $\rho = \sum_{i \in \widehat{S}} q_i$.
2. The total transmission rate of the SSs in $\widehat{S}$:
   $\rho = \sum_{i \in \widehat{S}} c_i$.
3. The summation of the queue lengths multiplying the transmission rate of each SS in $\widehat{S}$:
   $\rho = \sum_{i \in \widehat{S}} q_i \times c_i$.
4. The total transmission time of the SSs in $\widehat{S}$:
   $\rho = \sum_{i \in \widehat{S}} q_i / c_i$.
5. The summation of the complement of the hop count multiplying the queue lengths of each SS in $\widehat{S}$:
   $\rho = \sum_{i \in \widehat{S}} (h_{\max} - h_i + 1) \times q_i$, where $h_{\max}$ is the maximum hop count in the network.
6. The summation of the multiplication of the complement of the hop count, queue length, and transmission rate of each SS in $\widehat{S}$:
   $\rho = \sum_{i \in \widehat{S}} (h_{\max} - h_i + 1) \times q_i \times c_i$.

Then, for each criterion, the BS always selects the set of SSs with the largest ρ to serve in each iteration. Similar to the first strategy, the BS selects the minimum burst length, l_{min}, among these concurrent transmissions and allocates a burst with the length of l_{min} to each of them. For each burst, it is reserved one transmission overhead in front of it to avoid propagation delay. The above operations are repeated until the queues of all SSs become empty.

By allowing more concurrent transmissions, the scheduling length is reduced so that network throughput can be improved. However, since both strategies have to test all possible combinations of concurrent transmissions, the BS may encounter a high computation complexity. In addition, since the BS allocates the minimum burst length of all concurrent transmissions as the burst length of each transmission, some SSs may need to transmit their data using more than one bursts, which incur more transmission overheads. These issues will be addressed in the next section.

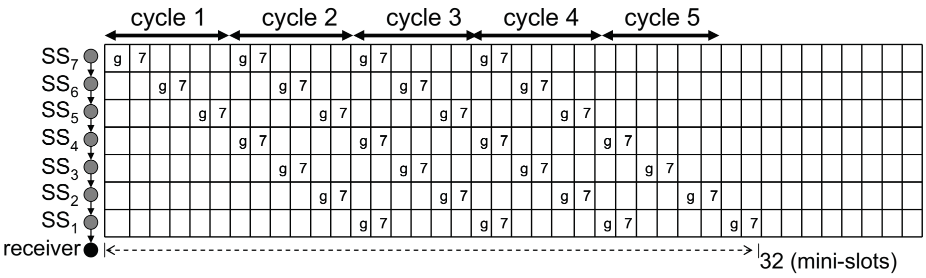
3.5. Regular Transmission Scheduling

The work of [9] aims at regular transmissions in grid-based WiMAX mesh networks, which have been deployed in many areas such as South Africa [6]. By employing regular transmissions, not only the scheduling complexity can be reduced, but also network performance can be improved by allowing more concurrent transmissions. The objective is to find out the optimal burst size for SSs to transmit so that the scheduling length (including the transmission overhead caused by guard time) can be minimized.

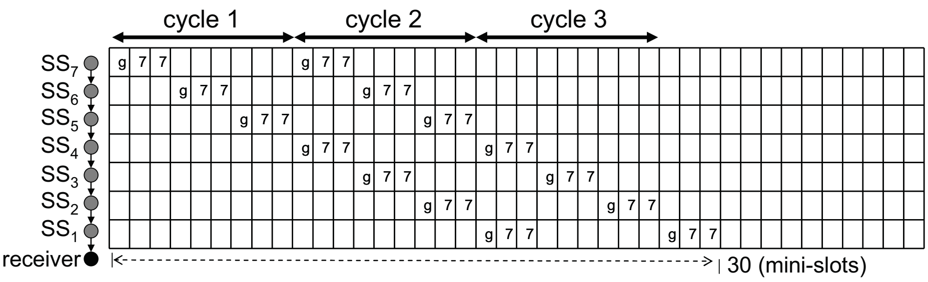
Given a grid-based WiMAX mesh network, the idea is to partition it into multiple chain-based networks and then schedule each chain. For each chain, there is only one SS which is responsible to collect data of each SS on the chain (we call it receiver). Then, the result can be extended to the whole grid-based network by letting the receiver in each chain to send data to the BS. For each chain, three possible cases may be considered:

- **There is only one traffic source and the receiver locates at one end of the chain.** This is the simplest case. Suppose that the interference range is fixed so that we can partition SSs into multiple disjointed groups to guarantee concurrent transmissions. In this case, the transmissions of SSs can be realized in a 'pipeline' manner, as shown in Fig. 5. Since all transmissions are regular, the problem is to find the optimal burst size to minimize the scheduling length. Below, we first show that different burst size will conduct different performance by Fig. 5(a) and (b). SS_7 is the traffic source with a request of four bytes. Assume that the guard time takes one minislot and the link rate is one byte per minislot. The interference range is two hops so that two SSs with a distance more than two hops can concurrently transmit their data without interfering with each other. In each cycle, three concurrent transmission flows can coexist: $SS_7 \rightarrow^{(1)} SS_6 \rightarrow^{(2)} SS_5 \rightarrow^{(3)} SS_4$, $SS_4 \rightarrow^{(1)} SS_3 \rightarrow^{(2)} SS_2 \rightarrow^{(3)} SS_1$, and $SS_1 \rightarrow^{(1)}$ receiver, where '$\rightarrow^{(i)}$' indicates the order of a transmission. In Fig. 5(a), the burst size is one minislot so that the cycle length is $[1\ (\text{guard time}) + 1\ (\text{burst size})] \times 3\ (\text{maximum hop count in a transmission flow}) = 6$ minislots. Since SS_7 has four-byte data and each burst can carry one-byte data, it takes totally $4/1 = 4$ cycles for SS_7 to send all its data to SS_4. In addition, SS_4 takes one cycle (that is, the fifth cycle) to send the last burst to SS_1 and SS_1 spends two minislots to forward this burst to the receiver. Therefore, the total scheduling length is $5\ (\text{the number of cycles}) \times 6\ (\text{cycle length}) + 2\ (SS_1 \text{ forwards the last burst}) = 32$ minislots. On the other hand, in Fig. 5(b), the burst size is two minislots so that each cycle takes $(1+2) \times 3 = 9$ minislots. Since SS_7 has four-byte data and each burst can carry two-byte data, it takes totally $4/2 = 2$ cycles to send all its data to SS_4. In addition, SS_4 takes one cycle (that is, the third cycle) to send the last burst to SS_1 and SS_1 spends three minislots to forward this burst to the receiver. Therefore, the total scheduling length is $3 \times 9 + 3 = 30$ minislots. It can be observed that the scheduling length can be reduced if the burst size is two minislots. The optimal burst size can be found using the similar calculation.

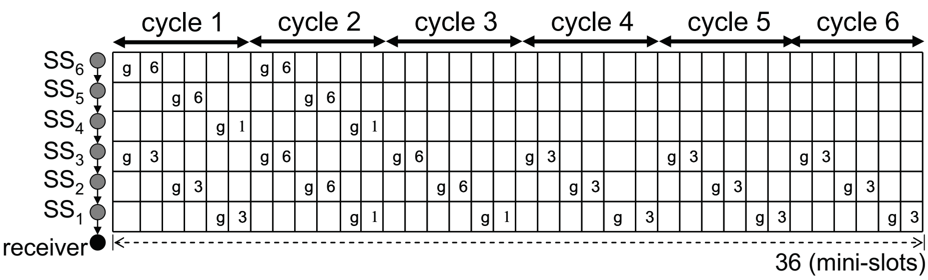
- **There are multiple traffic sources and the receiver locates at one end of the chain.** This case can be viewed as an extension of the previous case. Considering the same assumptions, Fig. 5(c) and (d) together give an example, where SS_6 and SS_3 has a



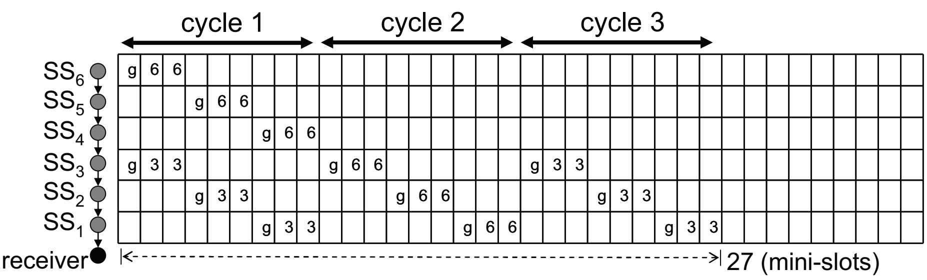
(a) One traffic source: SS_7 with a four-byte request



(b) One traffic source: SS_7 with a four-byte request



(c) Two traffic sources: SS_6 with a two-byte request and SS_3 with a four-byte request



(d) Two traffic sources: SS_6 with a two-byte request and SS_3 with a four-byte request

Figure 5. The case when the receiver locates at one end of the chain, where a minislot marked by 'g' is used for guard time and a minislot marked by a number i is used to transmit the data of SS_i. (a) The burst size is one minislot so that each cycle takes 6 minislots. The total scheduling length is 32 minislots. (b) The burst size is two minislots so that each cycle takes 9 minislots. The total scheduling length is 30 minislots. (c) The burst size is one minislot so that each cycle takes 6 minislots. The total scheduling length is 36 minislots. (d) The burst size is two minislots so that each cycle takes 9 minislots. The total scheduling length is 27 minislots.

request of two and four bytes, respectively. In each cycle, two concurrent transmission flows can coexist: $SS_6 \rightarrow^{(1)} SS_5 \rightarrow^{(2)} SS_4 \rightarrow^{(3)} SS_3$ and $SS_3 \rightarrow^{(1)} SS_2 \rightarrow^{(2)} SS_1 \rightarrow^{(3)}$ receiver. In Fig. 5(c), the burst size is one minislot so that the cycle length is $[1\ \text{(guard time)} + 1\ \text{(burst size)}] \times 3$ (maximum hop count in a transmission flow) $=$ 6 minislots. In the first two cycles, SS_6 can send all its data to SS_3. However, SS_3 can simultaneously send its one-byte data to the receiver after the first cycle. Thus, SS_3 has to send its remaining three-byte data and forward SS_6's one-bye data to the receiver, which take four extra cycles. Therefore, the total scheduling length is $[2+4\ \text{(the number of cycles)}] \times 6\ \text{(cycle length)} = 36$ minislots. On the other hand, in Fig. 5(d), the burst size is two minislots so that the cycle length is $(1+2) \times 3 = 9$ minislots. In the first cycle, not only SS_6 can send all its data to SS_3 but also SS_3 can send its two-byte data to the receiver. Thus, SS_3 requires only two extra cycles to send its two-byte data and forward SS_6's data to the receiver. Therefore, the total scheduling length is $(1+2) \times 9 = 27$ minislots. It can be observed that the scheduling length can be reduced if the burst size is two minislots. The optimal burst size can be derived following the similar calculation.

- **There are multiple traffic sources and the receiver does not locate at either end of the chain.** In this case, the chain can be separated into a *left subchain* and a *right subchain*. We can first calculate the number of groups of SSs that are allowed to concurrent transmit:

$$k = \begin{cases} H, & \text{if } 2 \le H \le 4 \\ 2H-4, & \text{if } H \ge 5 \end{cases},$$

where H is the minimum hop count that two SSs can concurrently transmit without interfering with each other. Then, from the end of each subchain, we first divide SSs into groups for concurrent transmission and then we 'shift' the groups of SSs by a number of Δ hops in the right subchain to avoid collision at the receiver, where $\Delta = 1$ if $H = 2$ and $\Delta = H-2$ if $H \ge 3$. Fig. 6(a) and (b) together give an example. In Fig. 6(a), assuming that $H = 3$, we have $k = H = 3$ groups of SSs (that is, G_0, G_1, and G_2). Since both SS_1 and SS'_1 will collide at the receiver, we need to shift the groups in the right subchain by a number of $\Delta = 1$ hop. On the other hand, in Fig. 6(b), assuming that $H = 5$, we have $k = 2H-4 = 6$ groups of SSs (that is, $G_0 \sim G_5$). Since both SS_1 and SS'_1 will collide at the receiver and SS_2 and SS'_2 will interfere with each other, we need to shift the groups in the right subchain by a number of $\Delta = H-2 = 5$ hops. Then, we can adopt the calculation in the previous case to find out the optimal burst size. Note that the scheduling length of the whole chain will be the maximum one of both the left and right subchains.

Then, the BS adopts a *fishbone-like routing* to collect data from all SSs. In particular, the network is formed by a number of *branch chains* and one *trunk chain*, where a branch chain is a vertical chain and the trunk chain is a horizontal chain containing the BS, as shown in Fig. 7. The intersected SS of a branch chain and the trunk chain will be the receiver in that branch chain. Two branch chains parallel with a distance more than or equal to H hops are allowed to concurrently transmit. After collecting all data along each branch

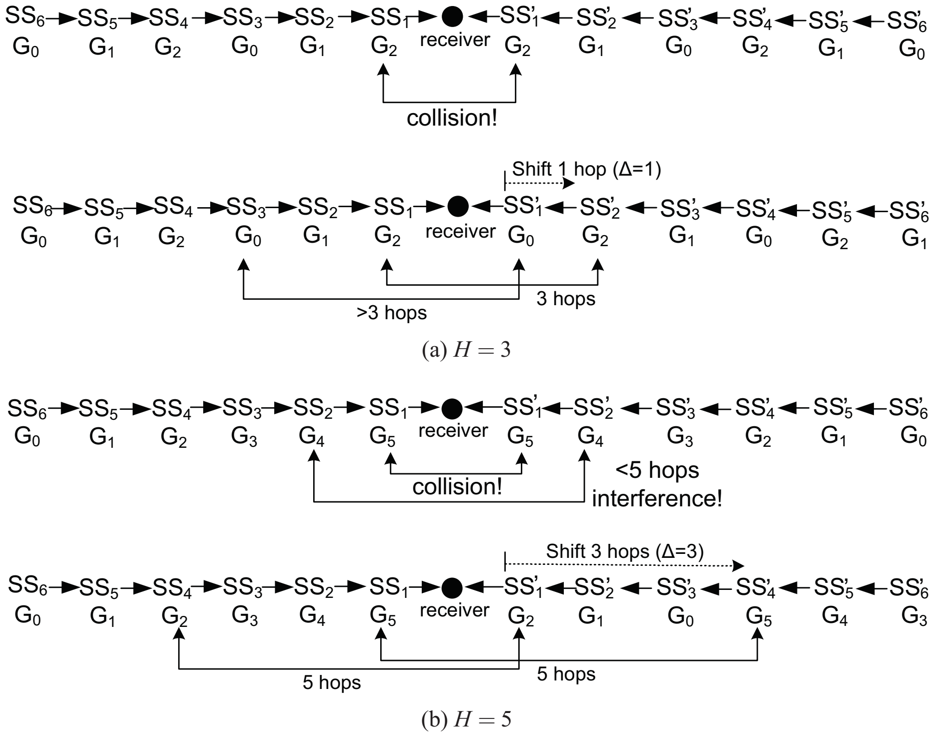


Figure 6. The case when the receiver does not locate at either end of the chain. (a) The chain requires $k = 3$ groups of SSs and the groups in the right subchain should be shifted by $\Delta = 1$ hop. (b) The chain requires $k = 6$ groups of SSs and the groups in the right subchain should be shifted by $\Delta = 3$ hops.

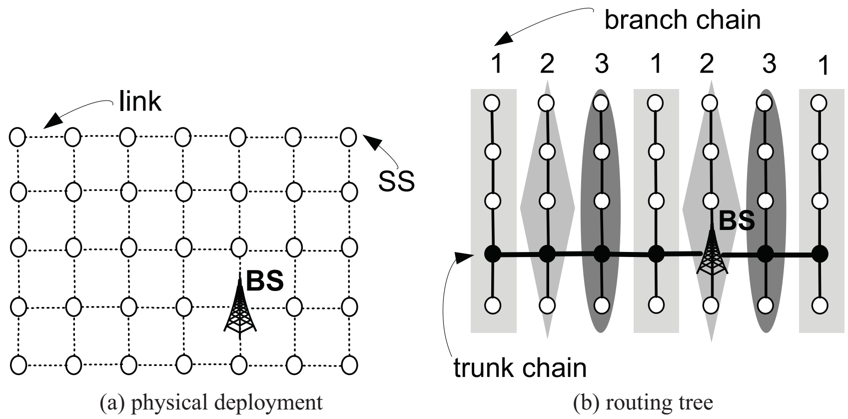


Figure 7. The fishbone-like routing scheme in a 5×7 grid-based WiMAX mesh network, where the branch chains with the same number are allowed to concurrently transmit.

node, the receivers (that is, the intersected SSs) will forward these data to the BS along the trunk chain. Fig. 7 give an example, where the branch chains with the same number are allowed to concurrently transmit.

By dividing a grid-based WiMAX mesh network into multiple chains and employing regular transmissions, not only the computation complexity of the scheduling solution can be reduced but also the overall guard time can be lowered down. Therefore, the scheduling length is reduced and network performance can be improved.

3.6. Multi-Channel Link Scheduling

The above studies assume that there is a single channel to be used. The work of [2] considers to exploit multiple channels with a single transceiver to eliminate the secondary interference[1] to further reduce the scheduling length. In particular, by using a multi-channel single transceiver MAC, the scheduler can assign additional channels for the interfered links to increase concurrent transmissions. This also makes the scheduling problem have to consider not only the link scheduling but also the channel assignment. Specifically, the scheduler will use an active link selection algorithm to determine the active links and their corresponding channels in each minislot. Given a routing tree and the bandwidth demand of each SS, the work of [2] expresses the scheduling problem as the link coloring problem on the routing tree to maximize the throughput under the condition of insufficient channels. Here, the links colored by the same color in a minislot means that they can transmit on the same channel without interference. The scheduler initially assigns each SS a *token* proportional to its bandwidth demand. In each minislot, the scheduler only considers those links with non-zero tokens and collects them to constitute the available link set. Then, it adopts the nearest selection algorithm to determine the colors of each link in the available link set. Finally, the tokens of the active links will be updated by increasing/decreasing one for the transmitter/receiver of the active links, respectively. Above operation is repeated until the tokens of all the links are zero. Here, we note that this work only focuses on the elimination of the secondary interference among links since the primary interference of links cannot be eliminated due to the mutual property of the single transceiver.

For each minislot, the nearest selection algorithm will select an uncolored link from the available link set iteratively and assign a color for the link that are different from the colors of the links causing secondary interference to it. This is shown in the literature that can lead a high performance. Below, we use an example to show how the scheduler colors the routing tree. We use Fig. 8 for example. We assume all SSs are with non-zero token. Since links (SS_1, BS) and (SS_2, BS) are the nearest to the BS (that is, one hop away from the BS), we choose the one with the smallest ID (that is, link (SS_1, BS)) and color it by color 1. Then, consider link (SS_2, BS), which is now the one nearest to the BS and with the smallest ID. Since link (SS_1, BS) causes the primary interference to it (at the receiver BS) and thus link (SS_2, BS) is omitted by the scheme and does not be colored. Now, consider links (SS_3, SS_1), (SS_4, SS_1), and (SS_5, SS_2), which have the same hop counts to the BS. Since links (SS_3, SS_1) and (SS_4, SS_1) cause the primary interference to link (SS_1, BS) at SS_1, they are omitted, too. On the other hand, since links (SS_5, SS_2) and (SS_1, BS) are interference-free, it is colored

[1] Here, we classify the interference into the primary one and the secondary one. For the primary interference, those links whose transmitters or receivers are the same as an active link's transmitter or receiver have the primary interference to that active link. For the secondary interference, those links (except for the links with the primary interference) whose transmitters or receivers are in the interference range of an active link have the secondary interference.

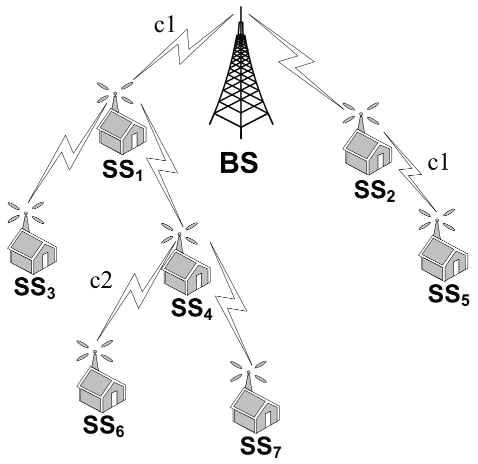


Figure 8. An example of link coloring by the nearest selection algorithm. Here, c_i is the color with index i. In this example, links (SS_1, BS) and (SS_5, SS_2) are colored by c_1. (SS_6, SS_4) is colored by c_2. The remaining links, such as links (SS_2, BS), (SS_3, SS_1), (SS_4, SS_1), (SS_7, SS_4), are not colored because they cause the primary interference to the colored ones.

by color 1 for reuse. Now, consider links (SS_6, SS_4) and (SS_7, SS_4), which are the ones with the minimal hop count to the BS in the available link set. We choose link (SS_6, SS_4) to color due to its smallest ID. Since it causes the secondary interference to link (SS_1, BS), it has to be colored by another color such as color 2. Therefore, there are three links and two colors (and thus two channels) to be scheduled in a minislot. That means that more links can be scheduled by exploiting multiple channels.

To further enhance the concurrent transmissions, the work of [10] adopts the *degree of saturation* [1] for link coloring. Given a routing tree, it transforms the edges in the routing tree to the vertices of a new graph, where two vertices have an edge if they interfere with each other (including the primary and secondary interferences). Then, this work defines the degree of saturation for a link by the number of neighbors with different colors and uses it to conduct the minimum coloring. When scheduling, a vertex with the highest saturation degree is selected and colored by the colors appearing before. This means that a new color is needed only when any used color can not be reused. These processes are repeated until all the vertices are colored.

The detailed operation of the scheme is performed as follows. First, it sets the saturation degrees of all vertices to zero. Then, it selects an uncolored vertex with the highest saturation degree. Here, the chosen vertex causing the primary interference to the colored ones is omitted. On the other hand, if more than one vortices have the same saturation degree, the scheme chooses the one having the largest number of uncolored neighbors. Next, the scheme colors the vertex with the colors used before (if possible). Finally, it updates the saturation degree of the uncolored vertices neighboring the colored vertex. Above steps are repeated until all vertices are colored.

By adopting the degree of saturation, more links can be activated and fewer channels can be used. This is more meaningful when the number of channels is insufficient.

Table 9. Comparison of the features for scheduling solutions in WiMAX networks.

work	methodology	spatial reuse	transmission order	batch allocation	transmission overhead	channel assignment
reference [5]	demand aggregation					
reference [7]	weight-based	✓				
reference [11]	traffic-aware	✓	✓			
reference [4]	token-based	✓	✓	✓		
reference [12]	maximal-concurrency	✓	✓	✓	✓	
reference [9]	regular transmission	✓	✓	✓	✓	
reference [2]	coloring (nearest selection)	✓	✓	✓		✓
reference [10]	coloring (satisfaction index)	✓	✓	✓		✓

4. Conclusion

WiMAX is developed to provide broadband wireless access. The WiMAX mesh network is defined to enhance the network coverage and provide a wireless backbone solution. In this chapter, we first introduce the features of WiMAX mesh networks in terms of access technology, network architecture, and the frame structure. Then, we provide a comprehensive survey of the scheduling problems and solutions in WiMAX mesh networks under different issues. For the single channel single transceiver, all the studies exploit the spatial reuse to improve the transmission efficiency. In particular, the studies of [4, 9, 11, 12] have took the transmission order into account to reduce the transmission delay. Further, the studies of [4,9,12] schedule the resource in a batch way, which can reduce the scheduling complexity. Moreover, both the studies of [9, 12] consider the transmission overhead incurred in the physical layer and employ spatial reuse to improve transmission efficiency. Additionally, the study of [9] considers the reduction of the transmission overhead based on the regular transmissions, which makes it more practical. On the other hand, the studies of [2, 10] considers to exploit multiple channels to further reduce the scheduling length. Different link coloring strategies are utilized. Table 9 gives a comparison of all the solutions which have been mentioned in this chapter.

References

[1] D. Brélaz, "New methods to color the vertices of a graph," *Communications of the ACM*, vol. 22, no. 4, pp. 251–256, 1979.

[2] P. Du, W.-J. Jia, L.-S. Huang, and W.-Y. Lu, "Centralized scheduling and channel assignment in multi-channel single-transceiver WiMAX mesh network," *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1734–1739, 2007.

[3] C. Eklund, R. B. Marks, K. L. Stanwood, and S. Wang, "IEEE standard 802.16: A technical overview of the wirelessMAN air interface for broadband wireless access," *IEEE Communications Magazine*, vol. 40, no. 6, pp. 97–107, 2002.

[4] B. Han, F.-P. Tso, L.-D. Lin, and W.-J. Jia, "Performance evaluation of scheduling in IEEE 802.16 based wireless mesh networks," *Computer Communications*, vol. 30, no. 4, pp. 782–792, 2007.

[5] IEEE Standard 802.16d-2004, "IEEE standard for local and metropolitan area networks part 16: Air interface for fixed broadband wireless access systems," 2004.

[6] D. Johnson, "Evaluation of a single radio rural mesh network in South Africa," *IEEE International Conference on Information and Communication Technologies and Development*, pp. 1–9, 2007.

[7] D. Kim and A. Ganz, "Fair and efficient multihop scheduling algorithm for IEEE 802.16 BWA systems," *International Conference on Broadband Networks*, pp. 833–839, 2005.

[8] B. Li, Y. Qin, C.-P. Low, and C.-L. Gwee, "A survey on mobile WiMAX," *IEEE Communications Magazine*, vol. 45, no. 12, pp. 70–75, 2007.

[9] J.-M. Liang, H.-C. Wu, J.-J. Chen, and Y.-C. Tseng, "Mini-Slot scheduling for IEEE 802.16d chain and grid mesh networks," *Computer Communications*, vol. 33, no. 17, pp. 2048–2056, 2010.

[10] Y.-L. Tang, R.-S. Cheng, T.-Y. Wu, and J.-P. Yu, "On centralized scheduling and channel assignment scheme for WiMAX mesh networks," *Journal of Internet Technology*, vol. 10, no. 3, pp. 207–215, 2009.

[11] H.-Y. Wei, S. Ganguly, R. Izmailov, and Z. J. Haas, "Interference-aware IEEE 802.16 WiMAX mesh networks," *IEEE Vehicular Technology Conference (VTC)*, pp. 3102–3106, 2005.

[12] J. Zhang, H.-L. Hu, L. Rong, and H.-H. Chen, "Cross-Layer scheduling algorithms for IEEE 802.16 based wireless mesh networks," *Wireless Personal Communications*, vol. 51, no. 3, pp. 615–634, 2009.