

A Novel MAC-layer Scheduling Scheme
for Improving soft-QoS Support
in IEEE 802.16(d) Mesh Networks

Student: Ku-Han Fang

Advisor: Prof. Shie-Yuan Wang



A Thesis

Submitted to Institute of Network Engineering

College of Computer Science

National Chiao-Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

Hsinchu, Taiwan

June 2006

A Novel MAC-layer Scheduling Scheme for Improving QoS Support in IEEE 802.16(d) Mesh Networks

Student: Ku-Han Fang

Advisor: Prof. Shie-Yuan Wang

Institute of Network Engineering
National Chiao-Tung University

ABSTRACT

The IEEE 802.16(d) standard (also known as WiMAX) is a promising technology for future fixed broadband wireless access (FBWA) systems. There are two operation modes defined in this standard. First, the point-to-multipoint (PMP) mode aims to replace the traditional wired last-mile solutions. Second, the Mesh mode is designed for the next-generation wireless metropolitan area networks (Wireless-MANs). In the 802.16 Mesh mode, network resource allocation can be handled in the centralized and distributed scheduling modes.

In the distributed scheduling mode, the minislot utilization is essential to achieving good performances of MAC-layer scheduling. In this thesis, we propose a novel MAC-layer scheduling scheme, named the “*slicing-based scheme*,” to improve network utilization. Besides, it can be used to provide better soft-QoS support. Both the multi-grant and multi-request mechanisms in this scheme are proposed and their performances are compared against those of the original scheme.

Our simulation results show that the slicing-based scheme significantly increases the MAC-layer minislot utilization. Thus, it generates good application performances and decreases the packet delay. In addition, our proposed scheme can better provide soft-QoS support.

Keywords: mesh networks, scheduling, QoS, IEEE 802.16, WiMAX.

Contents

Abstract	i
Contents	ii
List of Figures	iv
List of Tables	vi
1 Introduction	1
2 Related Work	3
3 Background	4
3.1 MAC-layer Mesh Operations	5
3.1.1 Basic Technologies	5
3.1.2 Network Entry Process	10
3.1.3 Link Establishment Process	12
3.1.4 Network Synchronization	12
3.1.5 Distributed Scheduling	15
3.2 PHY-layer Mesh Operations	18
4 Slicing-based Scheduling Scheme	20
4.1 Original Scheduling Scheme	20
4.2 Proposed Scheduling Scheme	23
4.2.1 Slicing-based scheme with multi-grant mechanism (SMG)	24



4.2.2	Slicing-based scheme with multi-request mechanism (SMR) . . .	28
4.3	Applications to soft-QoS Support	34
5	Functionality Validation	36
5.1	Corresponding Throughputs	36
5.1.1	Permanent allocation policy	37
5.1.2	On-demand allocation policy	40
5.2	Slicing-based Scheduling Scheme	44
5.2.1	SMG Scheme	45
5.2.2	SMR Scheme	46
5.3	The Soft-QoS Support	47
6	Performance Evaluation	49
6.1	Simulation Results on The SMG Scheme	49
6.1.1	Used Network Configuration	50
6.1.2	Minislot Utilization and Application Throughput	51
6.1.3	Network Scalability Issue	56
6.1.4	Availability Slot Size Effects	58
6.2	Simulation Results on Soft-QoS Support	60
6.2.1	Used Network Configuration	60
6.2.2	End-to-End Traffic	61
6.2.3	Bandwidth Reservation	63
7	Future Work	66
8	Conclusion	67
	Bibliography	69

List of Figures

3.1	The IEEE Std. 802.16 protocol layering	5
3.2	Mesh CID construction: (a) Broadcast CID and (b) Unicast CID . . .	6
3.3	The MAC PDU format	7
3.4	The Mesh frame structure	8
3.5	Mesh subframes in detail	9
3.6	The Registration Process	11
3.7	The procedure of the Link Establishment Process	13
3.8	The Next Xmt Time in both sender's and neighbors' views	14
3.9	The message flows of centralized scheduling and distributed scheduling	16
3.10	The three-way handshake mechanism to establish a schedule	17
3.11	Channel coding scheme	18
4.1	The required time for a three-way handshake procedure	21
4.2	The difference of granting resources between using (a) the original scheme and (b) the SMG scheme	26
4.3	The difference of required three-way handshake procedures between using (a) the original scheme and (b) the SMG scheme	27
4.4	The different system architectures for (a) the original scheme, (b) the original scheme using multi-connection mechanism, and (c) the SMR scheme	29
4.5	The different IE formats: (a) the original and modified Request IE formats and (b) the original and modified Grant IE formats	32

4.6	The difference of CID format between the original one and the modified one	33
4.7	The system architecture when the QoS mechanism is applied	35
5.1	The required TxOpps in the schedule on-demand process (a) in the best case and (b) in the worst case	41
5.2	The overhead of wasted frames (a) in the best case and (b) in the worst case	44
5.3	The difference of granting resources between using (a) the original scheme and (b) the SMG scheme with granting threshold value of 3	45
5.4	The difference of requesting resources between using (a) the original scheme and (b) the SMR scheme	47
6.1	(a) a chain network consisting of 21 nodes and (b) a 5x5 grid network used in Section 6.1	51
6.2	Minislot utilization vs. grid network scale	57
6.3	Network capacity vs. grid network scale	57
6.4	The effects of availability slot size on the number of a node's granting minislots per frame	59
6.5	The effects of availability slot size on application performance	59
6.6	The network configuration used in Section 6.2	61

List of Tables

3.1	Mac Management messages used in the Mesh mode	8
3.2	Mandatory PHY modes	19
5.1	The used simulation parameters in Chapter 5	37
5.2	Theoretical MAC-layer throughput and realistic application throughput using mandatory PHY modes	39
5.3	The parameters used in on-demand allocation policy	40
5.4	A node's granting minislots per frame and application throughput of the SMG scheme with different granting threshold value and the original scheme	46
5.5	A node's requesting minislots per frame and application throughput using the original scheme and the SMR scheme	48
5.6	The requested bandwidth reservation and the obtained application throughput for different traffic flows	48
6.1	The used simulation parameters in Chapter 6	50
6.2	The performances of the SMG scheme with different granting threshold indices and the original scheme	54
6.3	MAC-layer and Application-layer performances when using four different schemes	64
6.4	Achievement ratio of requested bandwidth reservation for different traffic flows	65

Chapter 1

Introduction

The IEEE 802.16(d) standard [1] (also known as WiMAX) is a promising next-generation broadband technology. It defines the air interface of fixed broadband wireless access (FBWA) system for supporting multimedia services. The standard defines two operational modes: the Point-to-Multipoint (PMP) mode and the Mesh mode. The PMP mode is a novel last-mile technology to replace traditional wired solutions. In this mode, traffic only occurs between a base station (BS) and subscriber stations (SS). On the other hand, the Mesh mode is designed for constructing the infrastructure of wireless metropolitan area networks (Wireless-MANs). In this mode, traffic can occur directly between neighboring SSs and can be routed through SSs. *In this thesis, we only study the Mesh mode.*

In WiMAX Mesh networks, the standard defines two scheduling modes: the centralized scheduling mode and the distributed scheduling mode. There are many critical issues, such as QoS, resource utilization, and network performance, in these modes. Many researchers, however, discuss these issues by analytical methods [2]. Moreover, most of previous work only focuses on the study of the centralized scheduling mode [4] [5]. As such, very few papers have studied the performances of the distributed scheduling mode. Over the NCTUns network simulator [6], we conducted a series of simulations and found two critical problems with the performances of the IEEE 802.16 Mesh networks. First, the MAC-layer minislot utilization is inefficient.

Second, the end-to-end traffic flows are served very poorly, even the traffic starvation problem may occur.

Motivated by the above observations, we design and implement a novel MAC-layer scheduling scheme, named the “*slicing-based scheme*.” This scheme, based on the distributed scheduling mode, is more efficient and flexible for bandwidth allocation between neighboring nodes. It has three main advantages. The first advantage is *enhancing the MAC-layer minislot utilization*. For the bandwidth request issued from the requesting node, the granting node can satisfy the request more aggressively and flexibly. Another advantage is *ensuring the fairness among traffic flows*. Based on the multi-connection mechanism, traffic flows are classified according to their traffic types. Then, the requesting node simultaneously issues the bandwidth requests for each classification of traffic flows. The last advantage is *improving the soft-QoS support*. We can better support the soft-QoS requirements than the original scheduling scheme.

Our contributions are threefold. First, we improve the application performances and reduce the packet delay because of the better MAC-layer minislot utilization. Second, the traffic starvation problem is solved by our proposed scheme. Third, our work can better provide soft-QoS support in terms of bandwidth reservation.

The rest of this thesis is organized as follows. The related work is discussed in Chapter 2. In Chapter 3, the background of the IEEE 802.16 Mesh mode is introduced. Our proposed scheduling scheme is explained in Chapter 4, and then its functionalities are validated in Chapter 5. In Chapter 6, we present the simulation results and performance evaluation. Finally, we propose possible extensions to our work in Chapter 7 and conclude the thesis in Chapter 8.

Chapter 2

Related Work

In [2], the authors analyzed the scheduler performance of the IEEE 802.16 Mesh networks. They developed a stochastic model for the distributed Mesh mode scheduler. With this model, the authors analyzed the scheduler performance under various conditions, and the analytical results were verified by the ns2 simulation results.

In [4] and [5], the work focuses on the study of the centralized scheduling mode and the routing issues. In the former, the authors investigated the spatial reuse in the IEEE 802.16 Mesh networks. They proposed a scheduling mechanism and a routing algorithm taking into account the interference in wireless environment to achieve maximum spatial reuse. In the latter, the authors designed a general algorithm for SSs to achieve concurrent transmission in both uplink and downlink streams. In addition, constructing and adjustment of routing tree is given in this paper.

Our work is rather different from the previous work, which mainly focuses on the distributed scheduling mode in the WiMAX Mesh networks. In this thesis, we identified two important problems in such a network. One is regarding the MAC-layer minislot utilization while another one is regarding soft-QoS support. We propose solutions to address these problems, and evaluate the effectiveness of these solutions by simulations.

Chapter 3

Background

IEEE 802.16(d) standard defines the specification of the air interface for FBWA systems. It is a promising next-generation broadband technology for supporting multimedia services. Both the MAC-layer and PHY-layer technologies are specified in this standard. The MAC layer is structured to support multiple PHY-layer specifications, each suited to a particular operational environment. In addition, the standard also supports IP-based network architecture (i.e. all WiMAX nodes are layer-3 devices), providing various services such as web browsing, real-time streaming and file sharing, etc.

The standard defines two operational modes: the PMP mode and the Mesh mode. Firstly, the PMP mode is designed to resolve the last-mile problem in traditional wired environment. When this mode operates in the 10-66 GHz licensed bands, its line-of-sight (LOS) applications offer data rates greater than 120 Mbit/sec. Secondly, the Mesh mode is an extension mode to support the infrastructure of Wireless-MANs. In addition, this mode operates in the licensed bands below 11 GHz and uses orthogonal frequency division multiplexing (OFDM) technology. Therefore, it can support non-LOS (NLOS) applications. Compared with the traditional 802.11-based mesh network, the WiMAX Mesh Network is able to provide higher throughput and larger coverage. *In this thesis, we only focus on the Mesh mode.*

Fig. 3.1 illustrates the reference mode defined in the standard. The MAC layer

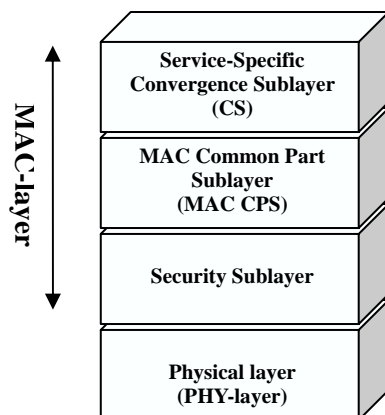


Figure 3.1: The IEEE Std. 802.16 protocol layering

consists of three sublayers, namely the Service-Specific Convergence Sublayer (CS), the MAC Common Part Sublayer (MAC CPS), and the Security Sublayer. The PHY layer adopts single-carrier technology to support LOS applications, and uses OFDM technology to support NLOS applications.

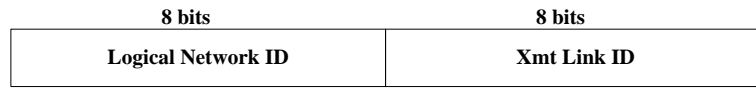
3.1 MAC-layer Mesh Operations

The MAC layer is composed of three sublayers: the CS, the MAC CPS, and the Security Sublayer. In the CS, the major function is classifying the data from upper layer, and then associating them to appropriate MAC connections. In the MAC CPS, all of the main Mesh operations, such as network entry process, network synchronization, and resources on-demand mechanism, etc., are specified. In the Security Sublayer, it provides the subscribers with privacy across FBWA system by encrypting the connections between BS and SS. *In this section, we regard the MAC CPS as MAC layer directly and only focus on this sublayer.*

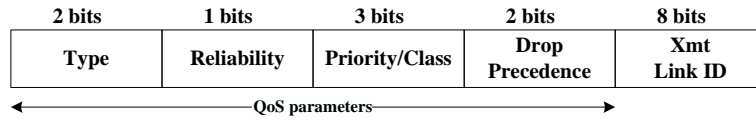
3.1.1 Basic Technologies

In the following, some terminologies used intensively in the standard are introduced.

- Mesh BS and Mesh SS



(a) Broadcast CID



(b) Unicast CID

Figure 3.2: Mesh CID construction: (a) Broadcast CID and (b) Unicast CID

The station that has a direct link to backhaul services (i.e., Internet) outside the WiMAX Mesh network is termed as a Mesh BS. Other stations are termed as Mesh SSs.

- Sponsoring Node, Candidate Node and Registration Node

These three types of nodes are Mesh nodes, and the terms only used in network entry process. The Mesh node assisting a new node in entering the WiMAX Mesh network in the network entry process is termed as a Sponsoring Node. The new node involved in this process is termed as a Candidate Node. The Mesh node responsible for assigning Node IDs to the new nodes is termed as a Registration Node.

- Link

A link is a directional mapping between two MAC peers for either control messages exchange or traffic flows transmission. It is identified by an 8-bit link identifier (Link ID) and used to construct the connection identifier (CID), as shown in Fig. 3.2. Note that a link is a logical communication between two Mesh nodes in their MAC-layer views.

- Connection

A connection, instead of a link, is a unidirectional mapping between two MAC peers. For a specified link, many different connections are constructed by

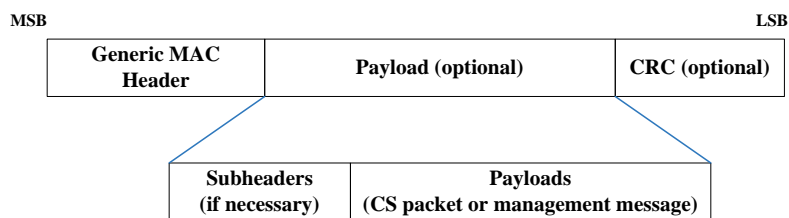


Figure 3.3: The MAC PDU format

either the various QoS parameters or the different network logical ID, i.e., *A link can construct multiple connections*. In addition, a connection is identified by a 16-bit connection identifier (CID). Note that a connection is a logical link between two Mesh nodes in their MAC-layer views as a link.

- Neighbor, neighborhood, and extended neighborhood

The station that a node has a direct link with is defined as the node's neighbor. Neighbors of a node form a neighborhood, and they are considered to be one-hop away from this node. An extended neighborhood covers not only a node's neighbors but also the neighbors of those being in the node's neighborhood.

In the following, we present the MAC Packet Data Unit (PDU) format and the frame structure.

MAC PDU

As shown in Fig. 3.3, the MAC PDU comprises a 6-byte generic MAC header, a variable-length Payload and an optional 4-byte cyclic redundancy check (CRC). In addition, the zero or more subheaders and the payloads are carried in the Payload. The payloads, instead of Payload, are used to carry either the data from upper layer (such as IP packet) or the MAC management messages.

The major functions of generic MAC header, subheader, and CRC are described as follows. The generic MAC header indicates what the subheaders are carried in the Payload and whether the CRC is appended. The extra information, such as the state of either payload fragmentation or payload packing, Xmt Node ID, is specified in the corresponding subheaders. The CRC covers the generic MAC header and

Table 3.1: Mac Management messages used in the Mesh mode

Message Name	Message Description
REG-REQ	Registration Request
REG-RSP	Registration Response
SBC-REQ	SS Basic Capability Request
SBC-RSP	SS Basic Capability Response
MSH-NCFG	Mesh Network Configuration
MSH-NENT	Mesh Network Entry
MSH-DSCH	Mesh Distributed Schedule
MSH-CSCH	Mesh Centralized Schedule
MSH-CSCF	Mesh Centralized Schedule Configuration

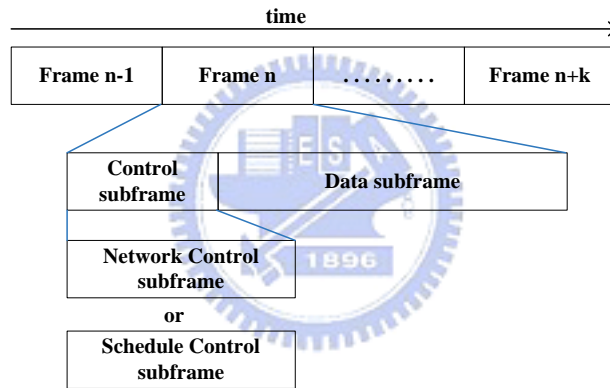
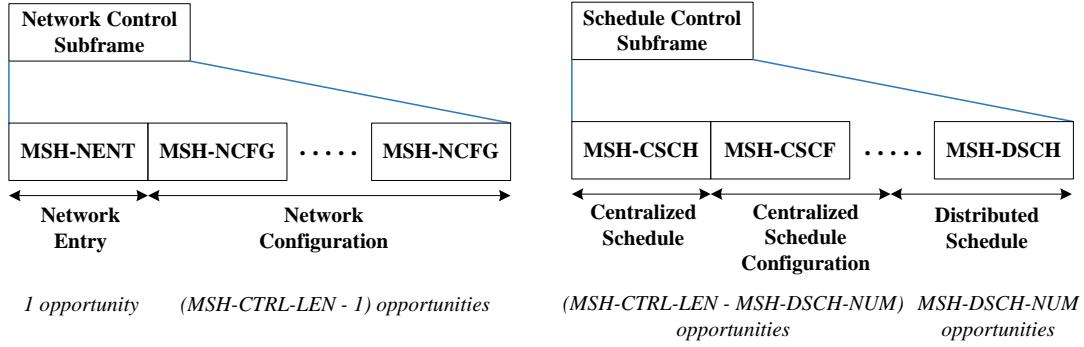


Figure 3.4: The Mesh frame structure

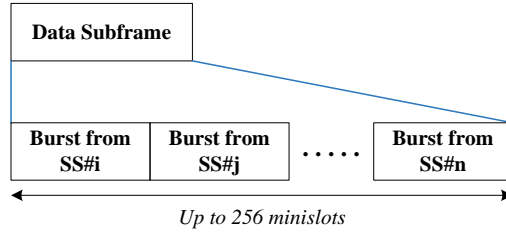
the Payload to achieve error-protecting. Table 3.1 lists the important management messages used in the Mesh mode.

Fragmentation is a process of dividing a MAC service data unit (SDU) into one or more MAC PDUs. This allows the efficient use of available bandwidth. The capabilities of fragmentation and reassembly are mandatory. On the contrary, packing is a process of packing multiple MAC SDUs into a single MAC PDU. The



(a) The network control subframe

(b) The schedule control subframe



(c) The data subframe

Figure 3.5: Mesh subframes in detail

capabilities of packing and unpacking are mandatory.

Frame Structure

The Mesh frame structure is illustrated in Fig. 3.4 and Fig. 3.5. Only time division duplex (TDD) is supported in the Mesh mode. A Mesh frame consists of a control subframe and a data subframe. In addition, the control subframe is divided into transmission opportunities (TxOpps); the data subframe is divided into minislots. There are two kinds of control subframes, namely the network control subframe and the schedule control subframe. The former is used for network configuration, new node entry, and network synchronization. The latter is used to exchange the coordinated scheduling information. In the following, we introduce the control subframe and data subframe in detail.

The length of a control subframe is a fixed value of $MSH-CTRL-LEN$ TxOpps. Each TxOpp comprises 7 OFDM symbols. A control subframe is either a network

control subframe or a schedule control subframe. In the network control subframe, one TxOpp is reserved for the network entry process, and the (MSH-CTRL-LEN - 1) TxOpps following the reserved one is used for the network configuration. On the other hand, in the schedule control subframe, MSH-DSCH-NUM defines the number of MSH-DSCH TxOpps per schedule control subframe. In addition, during this subframe, the first (MSH-CTRL-LEN - MSH-DSCH-NUM) TxOpps are reserved for centralized scheduling. The remainder is allocated for distributed scheduling. The parameters about the Mesh frame format are carried in the Network Descriptor of NCFG message.

The data subframe is divided into (up to) 256 minislots. In this subframe, the coordinated scheduling data and the uncoordinated scheduling packets will take place. A scheduled allocation consists of one or more minislots.

3.1.2 Network Entry Process

By the Network Entry Process, a new node (for Mesh SS only) can attach a Mesh network. Upon finishing this process, the node is able to start scheduled transmission. In the following, we elaborate this procedure step by step.

1. The new node first listens to the ongoing transmissions in the air, searching for MSH-NCFG messages to synchronize coarsely with the Mesh network. In the meantime, the new node shall build a physical neighbor list according to the information carried in the MSH-NCFG message.
2. When enough information is acquired, the new node shall select a Sponsor Node from its neighbor list, and it becomes a Candidate Node. Moreover, the new node shall synchronize its time to this Sponsor Node assuming 0 propagation delay.
3. By exchanging MSH-NENT and MSH-NCFG messages, the Candidate node and Sponsor node establish a temporary schedule, named Sponsor Channel. From that moment on, activities between the two peers are over this Channel.

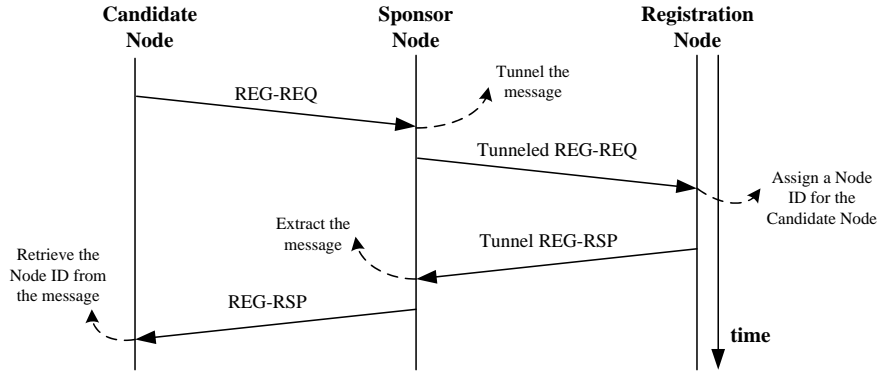


Figure 3.6: The Registration Process

4. Through the Sponsor Channel, the Sponsor Node assists the Candidate node to finish the basic capabilities negotiation and authorization with the BS. Afterward the Candidate node shall perform the Registration process to obtain its Node ID, as shown in Fig. 3.6.
5. At the beginning, the Candidate Node transmits a REG-REQ message to register with the Registration Node via its Sponsor Node. Upon receiving the REG-REQ message the Sponsor Node shall tunnel the message by prepending a tunnel subheader, a UDP header, and an IP header. The tunneled message is sent to the Registration Node, which can optionally be co-located with the Mesh BS. Upon the reception of this message, the Registration Node assigns the Node ID of the Candidate Node, replying tunneled REQ-RSP message. When receiving this tunneled message, the Sponsor Node shall extract the message and forward the REQ-RSP message to the Candidate Node. Eventually, the Candidate Node obtains its Node ID assigned.
6. Upon finishing the Registration process, the Candidate Node continues to acquire an IP address using DHCP, retrieve the current system time via the protocol defined in IETF RFC 868, and download the file containing operational parameters using TFTP.
7. Finally, upon completing above processes, the Candidate Node closes the Spon-

sor Channel and becomes a functional node.

3.1.3 Link Establishment Process

The Link Establishment Process is a three-way handshake procedure that performs simple authentication and establishes two unidirectional links between a pair of neighboring functional nodes. Before the new node can transmit data packets, it has to establish links with its neighbors by following this procedure. In the following, we take an example to explain this procedure step by step and illustrate it in Fig. 3.7.

1. Node 1 first initials the Link Establishment Process by sending a challenge message to Node 2.
2. Upon receiving the challenge message, the Node 2 first authenticates Node 1. If the authentication is successful, the Node 2 will create a link from itself to Node 1, and then send back a response challenge message carrying this link value to Node 1.
3. Similarly, on receiving the response challenge message of Node 2, the Node 1 first performs the authentication for Node 2. If the authentication is successful, a link is established from itself to Node 2. Finally, the Node 1 will transmit an acceptance message carrying the link value to Node 2, which indicates the Link Establish Process has been finished. Therefore, two one-way links are created safely between the Node 1 and Node 2.

3.1.4 Network Synchronization

In the Mesh network, functional nodes periodically broadcast MSH-NCFG messages to exchange network configuration information with their neighbors, which named the Network Synchronization. Moreover, the transmission timing of the synchronized messages is determined by a pseudo-random algorithm. The algorithm works without explicit negotiation and is completely distributed, fair, and robust. In the

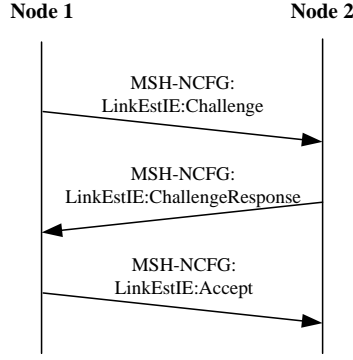


Figure 3.7: The procedure of the Link Establishment Process

following, we introduce the Network Synchronization and the pseudo-random algorithm respectively.

In the Network Synchronization, a node does not broadcast its exact next **Xmt Time** (i.e., the time slot when a node transmits its MSH-NCFG messages), but advertises its neighbors of an interval covering this actual **Xmt Time**. The interval is only composed of a 5-bit **Next Xmt Mx** and a 3-bit **Xmt Holdoff Exponent** for reducing the signaling overhead. Consequently, as shown in Fig. 3.8, the sender (i.e., the node transmitting its MSH-NCFG messages) knows its next transmit time (**Next Xmt Time**) very clearly, however, its neighbors only know the interval that covers the actual **Next Xmt Time** of the sender. The interval of the **Next Xmt Time** can be computed as follows:

$$2^{XmtHoldoffExponent} \cdot NextXmtMx < NextXmtTime \leq 2^{XmtHoldoffExponent} \cdot (NextXmtMx+1)$$

A node broadcasts not only its own **Next Xmt Mx** and **Xmt Holdoff Exponent** but also these two values of all its one-hop neighbors. Therefore, by this way, every regular node possesses the scheduling information within its extended neighborhood.

When a node is about to transmit the MSH-NCFG message, it shall schedule its next transmit timing of MSH-NCFG message. The determination of the **Next Xmt Time** is accomplished by a pseudo-random algorithm. According to the information deriving from its neighbors, the node can compute the **Next Xmt Time** using this algorithm. In the following, we define the terms used in the algorithm.

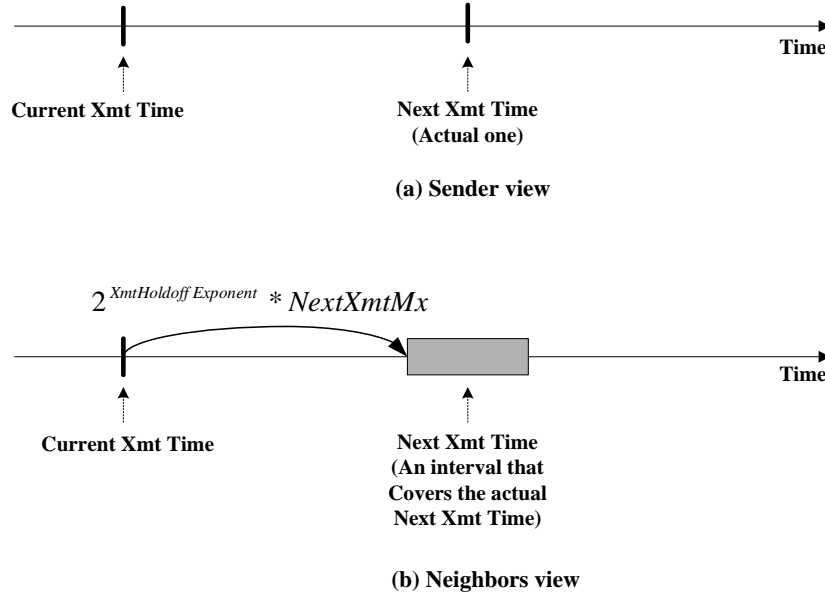


Figure 3.8: The Next Xmt Time in both sender's and neighbors' views

The **Xmt Holdoff Time** is the number of MSH-NCFG TxOpps after **Next Xmt Time**. In this period, a node is not eligible to transmit any MSH-NCFG packet.

$$XmtHoldOffTime = 2^{XmtHoldoffExponent+4}$$

The **Earliest Subsequent Xmt Time** indicates the earliest time when a node is capable of competing the candidate TxOpp, and can be obtained by

$$EarliestSubsequentXmtTime = 2^{XmtHoldoffExponent} \cdot NextXmtMx + XmtHoldoffTime + 1$$

By definition, the node only competes for TxOpps later than the current **Xmt Time** plus the node's **Xmt Holdoff Time**. That's when the **Temp Xmt Time** starts from. Neighbors are eligible to compete with the node for the **Temp Xmt Time** are those who meet one or more of the following conditions:

1. Its **Next Xmt Time** interval includes the **Temp Xmt Time**.
2. Its **Earliest Subsequent Xmt Time** is less than or equals to the **Temp Xmt Time**.
3. Its schedule is unknown.

A Mesh election for the **Temp Xmt Time** is held among the sender and its neighbors being eligible competing nodes. A pseudo-random mixing number $f(Node\ ID, Temp\ Xmt\ Time)$ is computed for each of the nodes involved in the election. If $f(Sender's\ Node\ ID, Temp\ Xmt\ Time)$ is the greatest among all numbers, the sender sets its **Next Xmt Time** equal to **Temp Xmt Time**, i.e., it can use the winning TxOpp to transmit its MSH-NCFG message. Otherwise, the **Temp Xmt Time** is advanced and the algorithm is repeated until the node wins the some TxOpp. Note that the fairness is ensured by the algorithm and the seeds (**Temp Xmt Time**) are different for every TxOpp.

After the **Next Xmt Time** is decided, it is converted into the corresponding **Next Xmt Mx** based on the specified **Xmt Holdoff Time**. Then the **Next Xmt Mx** and the specified **Xmt Holdoff Time** are added to the outgoing MSH-NCFG message.

3.1.5 Distributed Scheduling

In the Mesh network, the network resources (i.e., minislots) may be allocated by the three modes: the centralized scheduling mode, the distributed scheduling mode, and combination of both. These three modes also determinate the way data PDUs are routed in the network.

In the centralized scheduling mode, a Mesh network is partitioned into tree-based clusters. In each cluster, there is a Mesh BS acting as a centralized coordinator for allocating network resources to the Mesh Ss that it services. As shown in Fig. 3.9, the resource requests are issued to indicate the needed transmission and reception bandwidth, and then relayed by other nodes to the Mesh BS. The Mesh BS gathers all the resource requests issuing from its serving nodes in the bottom-up way, and then dispatches the granted schedules for each request in the top-down way. Finally, after determining the granted resources, the Mesh BS broadcasts the scheduling information, which are rebroadcasted by other nodes if needed.

On the other hand, in the distributed scheduling mode, schedules are established

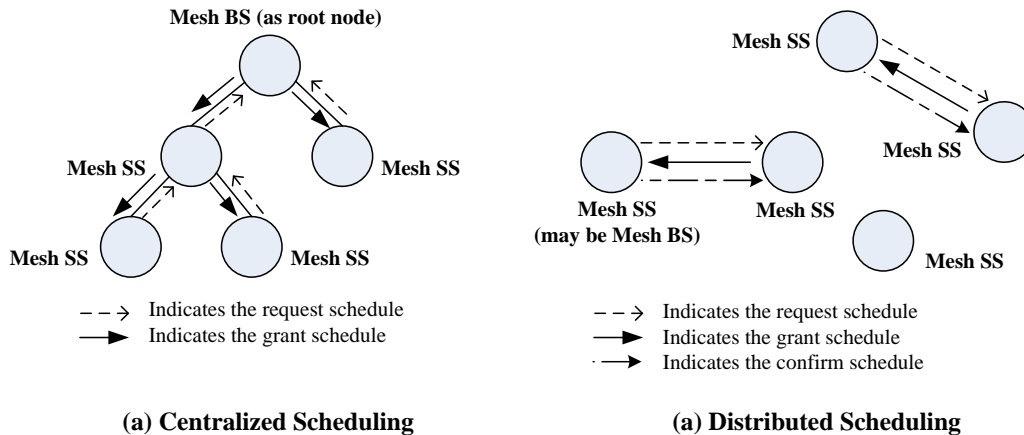


Figure 3.9: The message flows of centralized scheduling and distributed scheduling

in a distributed way in which all the Mesh nodes involved are regarded as peers (including the Mesh BS). As shown in Fig. 3.9, the schedules are set up by a three-way handshake mechanism, which is used to ensure the established schedules to be collision-free within the extended neighborhood. In the following we describe the distributed scheduling mode in more detail.

The distributed scheduling mode is divided into two operational modes, namely the coordinated mode and the uncoordinated mode. In both coordinated and uncoordinated modes, schedules are established between two nodes using a three-way handshake mechanism by the MSH-DSCH message exchange. In the coordinated distributed scheduling mode, the MSH-DSCH messages are transmitted over Tx-Opps in the schedule control subframe without collisions. In contrast, in the uncoordinated distributed scheduling mode, the MSH-DSCH messages can only be exchanged in the data subframe, i.e., collisions may occur. *In this thesis, we only focus on the coordinated distributed scheduling mode.*

In coordinated distributed scheduling mode, all nodes including Mesh BS transmit the MSH-DSCH message periodically in the control subframe to announce their schedules. *The transmission timing is determined by the same algorithm used for MSH-NCFG messages.* Therefore, the resulting transmissions of the MSH-DSCH

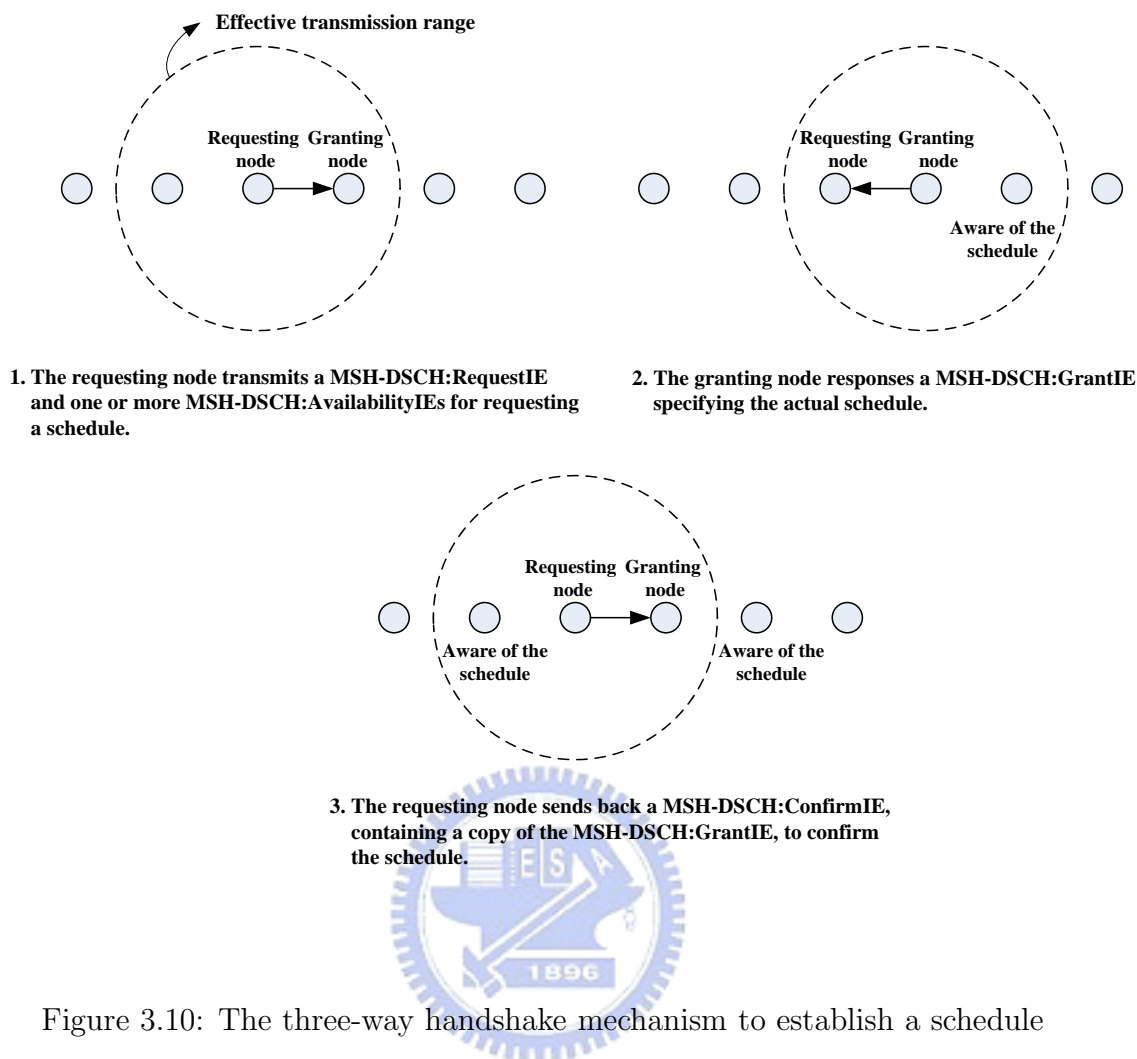


Figure 3.10: The three-way handshake mechanism to establish a schedule

messages are collision-free.

There are four kinds of information elements (IEs) that can be included in a MSH-DSCH message. The **MSH-DSCH:SchedulingIE** carries the coordinated distributed scheduling information: **Next Xmt Mx** and **Xmt Holdoff Exponent**. Each Mesh node shall broadcast these two information of its own and all its one-hop neighbors in every its MSH-DSCH TxOpp. The **MSH-DSCH:RequestIE** is used to convey resource requests on a specified link with the demand expressed in minislots. The **MSH-DSCH:AvailabilityIE** indicates free minislots ranges of the requesting node, which one **MSH-DSCH:RequestIE** can be corresponded to multiple **MSH-DSCH:AvailabilityIEs**. The granting node uses the **MSH-DSCH:GrantIE** to specify the range of granted minislots which selected from the free minislots reported by the requesting node. When sent by the requesting node, the **MSH-DSCH:GrantIE**

acts as a grant confirmation, i.e., MSH-DSCH:ConfirmIE.

In the following, we explain the three-way handshake mechanism step by step, which illustrated in Fig. 3.10. Note that each MSH-DSCH message transmission is just requiring one TxOpp. Therefore, it needs three time TxOpps to achieve the three-way handshake mechanism, which two are for requesting node, and one is for granting node.

1. The requesting node first transmits a MSH-DSCH:RequestIE and one or more MSH-DSCH:AvailabilitiIEs for requesting a schedule.
2. Upon reception of these messages, the granting node responds a MSH-DSCH:GrantIE specifying the actual schedule. In addition, the neighbors (except the requesting node) of the granting node shall assume that the schedule will take place as granted.
3. When receiving this message, the requesting node sends back a MSH-DSCH:ConfirmIE, containing a copy of the MSH-DSCH:GrantIE, to confirm the schedule to the granting node. Moreover, the third party (i.e., the requesting nodes neighbors except the granting node) shall keep silence during this schedule.

3.2 PHY-layer Mesh Operations

The physical layer for the Mesh mode is operating in the licensed bands below 11GHz and based on the OFDM technology. OFDM with a 256 point transform

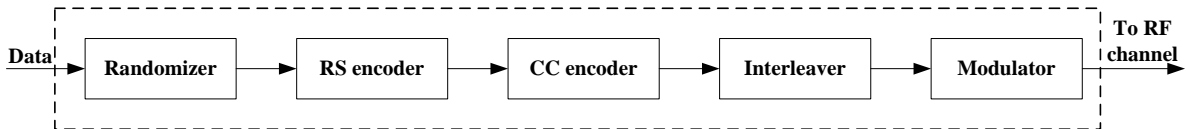


Figure 3.11: Channel coding scheme

Table 3.2: Mandatory PHY modes

Mandatory mode	Uncoded block size (bytes)	Coded block size (bytes)	RS code (N,K,T)	CC code rate
BPSK-1/2	12	24	(12,12,0)	1/2
QPSK-1/2	24	48	(32,24,4)	2/3
QPSK-3/4	36	48	(40,36,2)	5/6
16-QAM-1/2	48	96	(64,48,8)	2/3
16-QAM-3/4	72	96	(80,72,4)	5/6
64-QAM-2/3	96	144	(108,96,6)	3/4
64-QAM-3/4	108	144	(120,108,6)	5/6

is used to overcome delay spread, multipath, and inter-symbol interference (ISI) in this physical environment.

To better utilize the channel, a typical channel coding scheme is included in the standard, as shown in Fig. 3.11. The randomizer first scrambles the bit stream to avoid long runs of zeros or ones. The encoding is performed by passing the scrambled data blocks through the Reed-Solomon (RS) encoder and then passing the RS-encoded blocks through the convolutional code (CC) encoder. The RS code is a shortened and punctured code derived from a systematic $RS(N = 255, K = 239, T = 8)$ code using $GF(2^8)$. The CC is a punctured code derived from the basic CC 1/2. Various correcting capabilities can thus be realized by this concatenated coding scheme. The coded block is further interleaved to avoid long runs of bit errors. Finally, bits are entered serially to the constellation mapper.

Mandatory PHY modes are listed in Table 3.2 in the order of decreasing robustness (or increasing efficiency).

Chapter 4

Slicing-based Scheduling Scheme

In this chapter, we first describe the potential problems in the original scheduling scheme, and then the detailed design and implementation of the slicing-based scheduling scheme are presented elaborately. Finally, we discuss the soft-QoS support based on our proposed scheme.

4.1 Original Scheduling Scheme

In the distributed scheduling mode, data schedules are established between two peer nodes using a three-way handshake procedure, as mentioned in Section 3.1. This procedure requires three transmission opportunities to exchange three MSH-DSCH control messages (i.e., the request, the grant, and the confirm messages). By this way, it ensures these established schedules are collision-free.

A large delay, however, is adversely induced during a schedule establishment. As shown in Fig. 4.1, the required time for a three-way handshake procedure comprises the holdoff time and the contention time. The contention time is defined as the number of consecutive transmission opportunities in which a node should contend for access until it wins one. Thus, during a schedule establishment, the requesting node suffer from a delay of at least holdoff time plus contention time. This greatly reduces the network bandwidth utilization and increases the latency of the packets.

In the following, we discuss the three-way handshake procedure from two aspects,

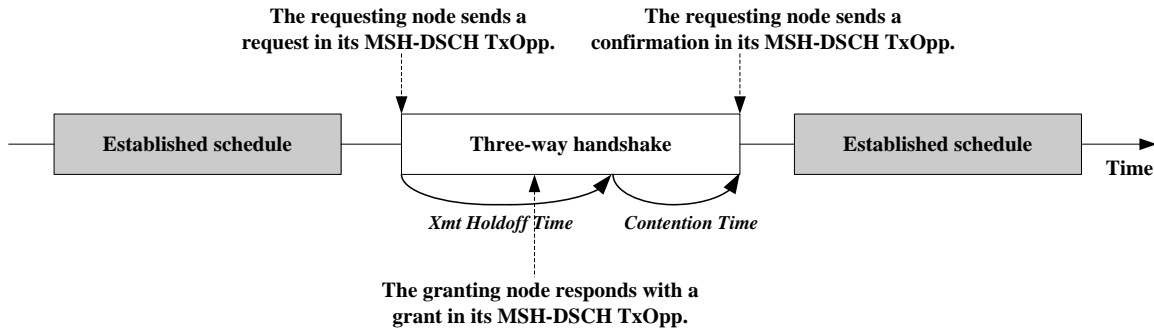


Figure 4.1: The required time for a three-way handshake procedure

which are the issuing grant mechanism and the issuing request mechanism, respectively. The potential problems attributed to these mechanisms are also described elaborately.

Issuing Grant Mechanism

In the original scheme, when the granting node receives a request, it assigns *just one schedule* for this request. Based on our design, this schedule must be a maximum allocation among the granting node's current available allocations. As such, during a three-way handshake procedure only one schedule can be established for a request between peer nodes. If the granting node cannot satisfy this request at a time, the requesting node has to establish more three-way handshake procedures to acquire its remaining bandwidth needed. Therefore, the network performances will decline. In the following, we list several serious problems in the original scheme.

1. Unexpected delay

As mentioned before, during a three-way handshake procedure, a large delay is experienced by peer nodes. In original scheme, the granting node can only assign one schedule for a request at a time. Thus, the requesting node may need to establish more three-way handshake procedures to acquire its remaining bandwidth needed. In such a condition, the packets may be transmitted in the current schedule, next schedule, or next next schedule and so on. Thus, these packets may suffer from a unexpected delay.

2. Inefficient minislot utilization

This problem is attributed to two reasons. First, since only one schedule can be established between peer nodes in a period, the network bandwidth are utilized incompletely. Second, in a highly-loaded traffic environment, it is easy to form many fragmented available resources. This is because many traffic flows compete for the networks resources in the same time and thus a verity of established schedules will be scattered in the network. As mentioned before, using the original scheme, the granting node can only assign one schedule at a time. Thus, these fragmented schedules cannot be utilized effectively.

3. Poor application performances

Since the performances of the data minislot utilization can be directly reflected in application performances. As such, in the original scheme, the application performances is poor due to inefficient minislot utilization.

Issuing Request Mechanism

In the original scheme, the requesting node inserts all kinds of traffic (e.g., TCP, greedy UDP, or real-time traffic) into the same connection queue and then issue a request for the mixed traffic in its MSH-DSCH TxOpp. As such, the end-to-end traffic flows between peer nodes are serviced very poorly, even the traffic starvation problem may occur. In the following, we list several problems in the original scheme.

1. Traffic starvation problem

As mentioned before, since a variety of traffic intensely competes for the network resources, some specified flows may not be served smoothly, TCP especially. As such, these flows may always not obtain the transmission minislots to sent their packets, and thus a traffic starvation problem may occur.

2. Packet long-delay problem

The traffic starvation problem may be addressed if the multi-connection mech-

anism are applied, which classifies the different traffic flows into the corresponding queues. Then the basic policy, round-robin, are used. The packets, however, may suffer from a large delay because these traffic flows take turns to establish their schedules.

4.2 Proposed Scheduling Scheme

In the Section 4.1, we discuss many potential problems, such as large packet delay time, inefficient resource utilization and traffic-starvation phenomenon, in the original scheme. Thus, in this thesis, we propose a more robust MAC-layer scheduling scheme, named the slicing-based scheduling scheme, to address above problems. This scheme, developed in the distributed scheduling mode, is more efficient and flexible for bandwidth allocation between neighboring nodes. It *slices* the available network resources (i.e., minislots) to achieve the goal that the schedules are established more tightly and efficiently. Therefore, the network resources can be more effectively utilized.

The Slicing-based scheme supplies two mechanisms: the multi-grant mechanism and the multi-request mechanism. Thus, there are three scheme combinations we can support, which are the slicing-based scheme with the multi-grant mechanism (SMG) scheme, the slicing-based scheme with the multi-request mechanism (SMR) scheme and the slicing-based scheme with the multi-grant and multi-request mechanisms (SMGR) scheme, respectively.

When applying the SMG scheme, the granting node can aggressively assign its available resources to satisfy the requesting node's resource requirements as possible as it can. This scheme exploits the available resources that may never be used in the original scheme, and thus the data minislot utilization can be significantly enhanced. On the other hand, when applying the SMR scheme, the requesting node can simultaneously issue at most four requests for each traffic flow in a MSH-DSCH TxOpp. Thus, the end-to-end traffic flows can be more smoothly served and the soft-QoS can be better supported.

In the following, we first introduce the slicing-based scheme with the multi-grant mechanism (SMG) in Section 4.2.1, and the slicing-based with the multi-request mechanism (SMR) is then described in the Section 4.2.2.

4.2.1 Slicing-based scheme with multi-grant mechanism (SMG)

As mentioned in Section 4.1, there are many significant problems, such as large packet delay and inefficient minislot utilization, in the original scheme. Thus, in the thesis, we propose a more robust scheme (i.e., the SMG scheme) to address these problems. In the following, we introduce this scheme in more details.

In the SMG scheme, when the granting node receives the bandwidth request issued from the requesting node, it first looks for the maximum allocation among its all of current available resources. If this requested resource is not filled up at the first granting process, the granting node will further seek for the available resource again. This process is repeated until (1) this request has been fulfilled, (2) the granting node has no any available resource to schedule and (3) the number of the appending grant IEs exceeds the maximum value defined in the standard (this maximum value is 63). After that, all of the grant IEs are encapsulated into a MSH-DSCH message, which each IE indicates a schedule used for data transmissions. This MSH-DSCH message is then transmitted in the granting node's next MSH-DSCH TxOpp. Thus, using this proposed SMG scheme, the granting node can issue multiple grants used for the same request in a MSH-DSCH TxOpp. The used algorithm is listed in the next page.

The requesting node, however, should reply the confirms, which each one is associated with one receiving grant, upon the reception of these resource grants. Thus, the requesting node can send multiple confirms in a MSH-DSCH TxOpp when using the SMG scheme. This proposed SMG scheme is compatible with the IEEE 802.16 standard because we use the fields which have been defined in the standard to achieve our scheme.

Fig. 4.2 shows an example representing the difference of granting resources be-

tween using the SMG scheme and using the original scheme. Assume that the requesting node demands a request which specifies the minislot range to 5 and the frame validity to 5 (i.e., this request can be treated as a 5x5 grid). In the original scheme, since only one schedule can be assigned at a time, the granting node assigns a 3x3 schedule. This is the maximum allocation among its all of current available resources. One sees that, however, many fragmented resources are not efficiently utilized. In contrast, in our proposed SMG scheme, more than one schedule can be assigned during a three-way handshake procedure (i.e., the 3x3, 3x2, and 2x2 schedules can be assigned at a time). As a result, using this SMG scheme, the network resources can be more efficiently utilized, and the MAC-layer performances can also be enhanced.

Algorithm 1 The SMG Scheme

```

1: if there is no pending request then
2:   do nothing
3: else
4:    $N \leftarrow$  the max number of grant IEs in a MSH-DSCH message
5:    $R \leftarrow$  the number of required minislots for the pending request
6:   for  $i = 1$  to  $N$  do
7:      $G_i \leftarrow$  the number of granted minislots for the  $i$ th grant
8:     if  $G_i = 0$  or  $R = 0$  then
9:       break
10:    end if
11:     $R \leftarrow R - G_i$ 
12:    append the  $i$ th grant IE with the  $G_i$  granted minislots
13:    to the outgoing MSH-DSCH message
14:  end for
15: end if

```

Based on above example, when using the original scheme, the requesting node has to further establish two three-way handshake procedures to obtain the remaining

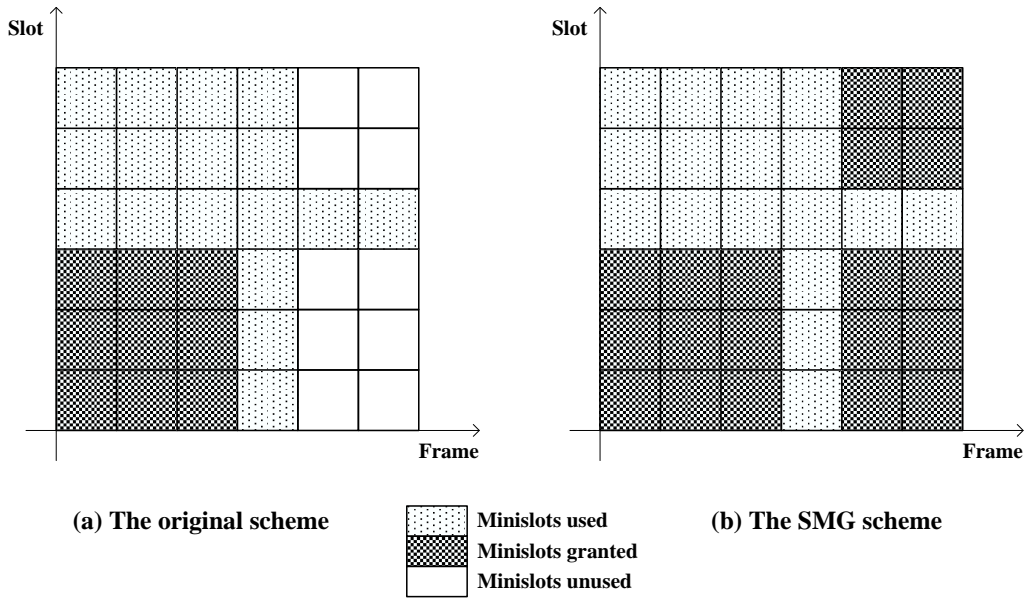


Figure 4.2: The difference of granting resources between using (a) the original scheme and (b) the SMG scheme

resources it needs. This is because the granting node cannot satisfy the requesting node's requirement in a three-way handshake procedure. In contrast, when using the SMG scheme, only one three-way handshake procedure is required. As such, we save unnecessary three-way handshake procedures. Thus, the packet delay can be significantly reduced because all of the required bandwidth are obtained in an on-demand procedure. The difference of required three-way handshake procedures between using the SMG scheme and the original scheme is illustrated in Fig. 4.3.

In the SMG scheme, we define a system parameter, named the granting threshold, G_{Thres} , which indicates that how many times of granting processes are performed for a request. Thus, the granting threshold can be used to limit the granting node to assign its resources in the SMG scheme. In a low-traffic-density environment, every node in the network should aggressively utilize the network resources as possible as they can. This is because it is hard to exhaust the network resources in such a network. Thus, the G_{Thres} value should be specified as large as possible. In contrast, in a high-traffic-density environment, the G_{Thres} value should be set to a small value to ensure the fairness of resource sharing. The effect of the G_{Thres} on the network performances are shown in Section 6.1.

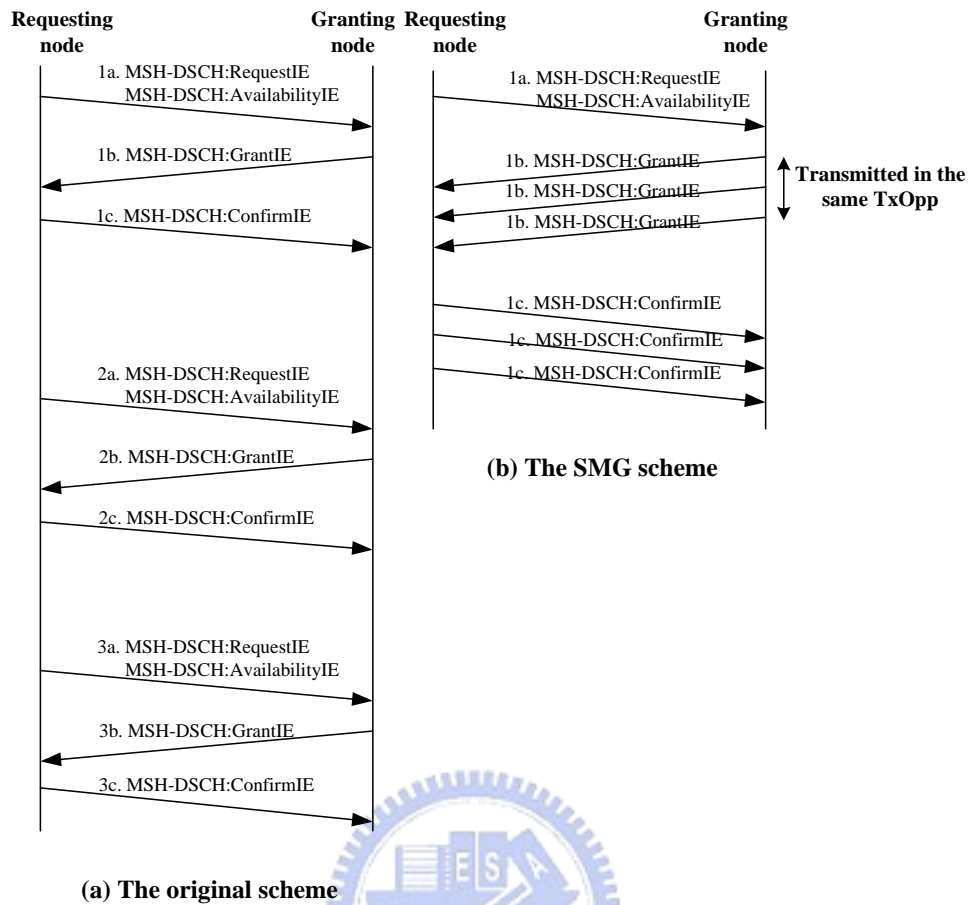


Figure 4.3: The difference of required three-way handshake procedures between using (a) the original scheme and (b) the SMG scheme

In the following, several advantages of the SMG scheme are listed.

1. Reduce packet delay

As mentioned before, the packet delay is significantly reduced because all the required bandwidth can be assigned as possible in a three-way handshake procedure.

2. Improve network performances

Using the SMG scheme, the data minislots are more efficiently utilized. Thus, the application performance can also be significantly enhanced.

3. Conform to standard

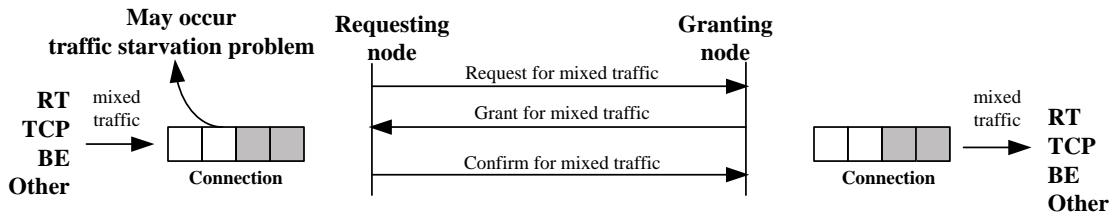
We use the fields which have been defined in the standard to achieve our proposed scheme. Therefore, the SMG scheme can be compatible with the IEEE 802.16 standard.

4.2.2 Slicing-based scheme with multi-request mechanism (SMR)

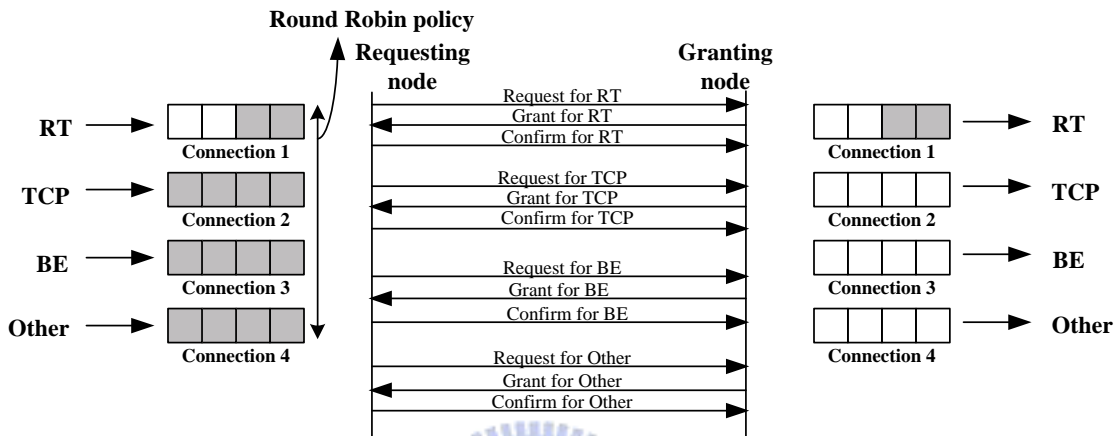
As mentioned in Section 4.1, using the original scheme, the end-to-end traffic flows are served very poorly, even the traffic starvation problem may occur. However, if a multi-connection mechanism is used and a round-robin policy is applied, the packets still may suffer from a large delay. Thus, in this thesis, we further propose the SMR scheme to solve these problems. In the following, we first explain why this proposed scheme is necessary and important. Then, the design and implementation of this SMR scheme are presented.

Fig. 4.4 shows the system architectures of the original scheme, the original scheme applying multi-connection mechanism, and the SMR scheme, respectively. In the original scheme, a variety of traffic flows are inserted into the same connection, and then the requesting node establishes a request specifying the requested bandwidth required by these mixed traffic flows. In a high-traffic-density environment, however, each traffic flow wants to contend for the network resources, and thus the traffic starvation problem may occur.

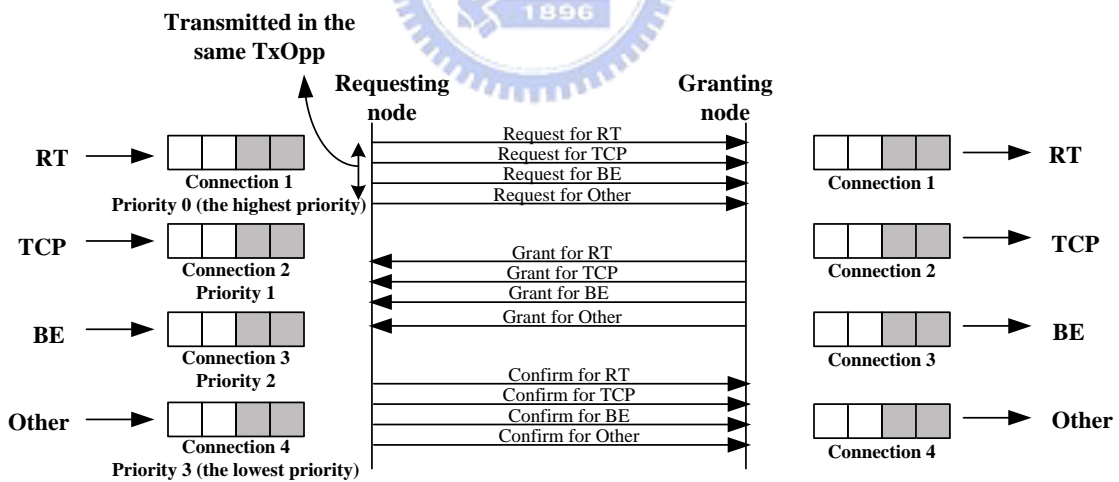
If the multi-connection mechanism is used, different traffic flows will be classified and inserted into different connection queues based on their traffic types. These connection queues are serviced in a round-robin manner. When a connection is serviced, the requesting node will issue a request for this connection. Using such a round-robin scheme, each connection may suffer from a large delay because it can be only served once in a round. A round is defined as the required time during which the requesting node establishes four data schedules, each of which is for a specific connection. Due to this reason, we propose the SMR scheme to address the above two problems.



(a) The original scheme



(b) The original scheme using multi-connection mechanism



(c) The SMR scheme

Figure 4.4: The different system architectures for (a) the original scheme, (b) the original scheme using multi-connection mechanism, and (c) the SMR scheme

In the SMR scheme, the multi-connection mechanism is also necessary for avoiding the traffic starvation problem. We classify the different traffic flows into four classes based on their traffic types. These four classes are TCP traffic (reliable, flow-controlled, congestion-controlled flows), best effort (BE) traffic, real-time traffic (RT), and other-type traffic. After being classified, each packet will be inserted into a connection queue based on its traffic type. Besides, we use a priority policy to decide the sequence of serving these traffic flows. In the following, we list the used algorithm and describe how the SMR scheme works.

Algorithm 2 The SMR Scheme

```

if there is no pending data in every connection then
2:   do nothing
   else
4:    $N \Leftarrow$  the number of connection
       $P \Leftarrow$  the priority count (is set to the value of highest priority by default)
6:   for  $i = 1$  to  $N$  do
       $C_{Selected} \Leftarrow$  the selected connection according to the value of  $P$ 
8:     if there is no pending data in the  $C_{Selected}$  then
        do nothing
10:    else
       $R_{Selected} \Leftarrow$  the number of requested minislots for the  $C_{Selected}$ 
12:      append a request IE with the  $R_{Selected}$  requested minislots
        to the outgoing MSH-DSCH message
14:    end if
       $P \Leftarrow P + 1$ 
16:  end for
end if

```

When a requesting node can transmit its MSH-DSCH message, it first checks whether any of its connections has pending data to transmit. If yes, these connections will be serviced based on their priorities. In other words, the requesting node

will first create a request for the connection with the highest priority among these connections. It then creates requests for the remaining connections in the order of their priorities until the requests for all of these connections have been created. Finally, it places those requests into the MSH-DSCH message to be sent. As such, at most four requests can be issued simultaneously in a MSH-DSCH TxOpp. Therefore, the requesting node can establish at most four schedules, each of which is for a respective connection, in a three-way handshake procedure.

As shown in Fig. 4.4 (a), in the original scheme, all kinds of traffic flows are inserted into the same connection queue. As such, the traffic starvation problem may occur, especially for TCP traffic. Besides, in a three-way handshake procedure, the requesting node can only establish a schedule, which is used to service the mixed traffic. Thus, some traffic flows may suffer from unexpected delay.

In Fig. 4.4 (b), the multi-connection mechanism is used to address the traffic starvation problem. Different traffic flows will be classified and inserted into different connection queues based on their traffic types. A round-robin policy is adopted to serve these traffic flows in turn and establish schedules for these traffic flows in turn. A large delay, however, may still be experienced by some traffic flows. This is because each traffic flow is served only once in a round-robin round.

As shown in Fig. 4.4 (c), in the SMR scheme, the multi-connection mechanism is also applied to solve the traffic starvation problem. Besides, using this scheme, the requesting node can establish at most four data schedules, each of which is for a specific traffic flow, in a three-way handshake procedure. As such, all of these traffic flows can obtain their required resources in every three-way handshake procedure. In the following, we present the required supports of the SMR scheme by the IEEE 802.16 standard.

As mentioned before, using the SMR scheme, the requesting node can issue distinct requests, each of which specifies the required bandwidth for a specific traffic flow, in a MSH-DSCH TxOpp. To support that, we exploit 2-bit space of the Link ID field in the MSH-DSCH:RequestIE message, as shown in Fig. 4.5 (a). Recall that a link is established between a node and its one-hop neighbor. As such, using the

(Original Request IE format)

8 bits	8 bits	3 bits	1 bit
Link ID	Demand Level	Demand Persistence	<i>reserved</i>

(Modified Request IE format)

2 bits	6 bits	8 bits	3 bits	1 bit
Flow Type	Link ID	Demand Level	Demand Persistence	<i>reserved</i>

(a) The original and modified Request IE formats

(Original Grant IE format)

8 bits	8 bits	8 bits	8 bits	1 bit	3 bits	4 bits
Link ID	Start Frame number	Minislot start	Minislot range	Direction	Persistence	Channel

(Modified Grant IE format)

2 bits	6 bits	8 bits	8 bits	8 bits	1 bit	3 bits	4 bits
Flow Type	Link ID	Start Frame number	Minislot start	Minislot range	Direction	Persistence	Channel

(b) The original and modified Grant IE formats

Figure 4.5: The different IE formats: (a) the original and modified Request IE formats and (b) the original and modified Grant IE formats

SMR scheme, a node can have at most 63 neighbors in its one-hop neighborhood. This is acceptable in the IEEE 802.16 Mesh networks. If a node has a large number of nodes (e.g., 64 or higher) in its one-hop neighborhood, the network bandwidth may be insufficient for a node’s and its neighbors’ requires. Thus, the application performances may not be good in such a environment. Using the SMR scheme, the requesting node can assign a traffic flow type in the extended “Flow Type” field. This extended field indicates that which traffic flow a request specifies. And then, these distinct requests will be encapsulated into a MSH-DSCH message to be transmitted.

Upon the reception of such a MSH-DSCH message, the granting node assigns the data schedules, each of which is for a specific request (i.e., for a specific traffic flow). To support that, we also exploit 2-bit space of the Link ID field in the MSH-DSCH:GrantIE message, as shown in Fig. 4.5 (b).

In the SMR scheme, we use four Mesh connections to service different traffic

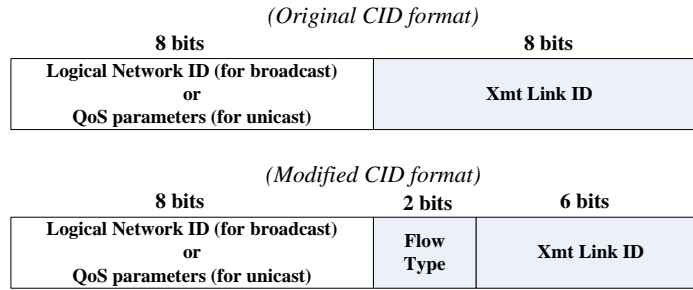


Figure 4.6: The difference of CID format between the original one and the modified one

flows. As such, we should have an ability to identify which traffic flow a Mesh connection serves. To support that, the CID format should be modified as the Request IE and GrantIE. As shown in Fig. 4.6, ones see that the 2-bit Xmt Link ID field is also used to indicate which traffic flow the connection serves. Note that it should be the 0X3F, instead of 0XFF, to indicate the value of MAC management broadcast now. This is because only the 6-bit Xmt Link ID field can be used.

A priority policy is used to determine which connection should be serviced first. In other words, the requesting node first creates a request for the connection with highest priority among these connections, and then appends this request into a outgoing MSH-DSCH message. The remaining connections are then served in the order of their priorities, and these created requests are also appended into this MSH-DSCH message. The connection serving sequence may have a great effect on the traffic flow throughputs. This is because the granting node first schedules its resources to the request, which is encapsulated at the head of the incoming MSH-DSCH message. Then, the remaining requests are scheduled in the order of their appending sequence in this MSH-DSCH message. Thus, after many schedules are assigned to those preceding requests, only a few remaining resources can be provided for these remaining requests that have not been serviced yet.

In the following, several advantages of the SMR scheme are listed.

1. Solve traffic starvation problem

As mentioned before, in the SMR scheme, the multi-connection mechanism is used to classify and insert different traffic flows into different connection

queues. As a result, the traffic starvation problem can be solved.

2. Reduce packet delay

In the SMR scheme, the requesting node can simultaneously establish at most four schedules, each of which is for a specific traffic flow, during a three-way handshake procedure. As such, all of these traffic flows can obtain their required bandwidth in every three-way handshake procedure. Therefore, the packet delay can be significantly reduced.

3. Improve the soft-QoS support

Using the SMR scheme, the end-to-end traffic flows can be served more smoothly and the soft-QoS can be better supported. This will be discussed in Section 4.3 in more details.

4.3 Applications to soft-QoS Support

In the original scheme, the traffic flows between two peer nodes are served very poorly, even the traffic starvation problem may occur. In addition, the QoS requirements (e.g. bandwidth reservation) cannot be supported well using this original scheme. In fact, all of these problems are the soft-QoS critical issues. Thus, in this section, we propose a SMGR scheme to address these problems. As mentioned in Section 4.2, the SMGR scheme consists of two schemes: the SMG scheme and the SMR scheme. Using the SMGR scheme, the end-to-end traffic flows can be served smoothly and soft-QoS can be better supported. In the following, we explain why the SMGR scheme is required and important from two aspects, which are the SMG scheme and the SMR scheme.

As mentioned in Section 4.2.2, the SMR scheme uses the multi-connection mechanism to classify and insert different traffic flows into different connections. Thus, the traffic starvation problem can be avoided. Besides, the requesting node can simultaneously establish at most four schedules, each of which is for a specific traffic flow, during a three-way handshake procedure. In other words, all of these traffic

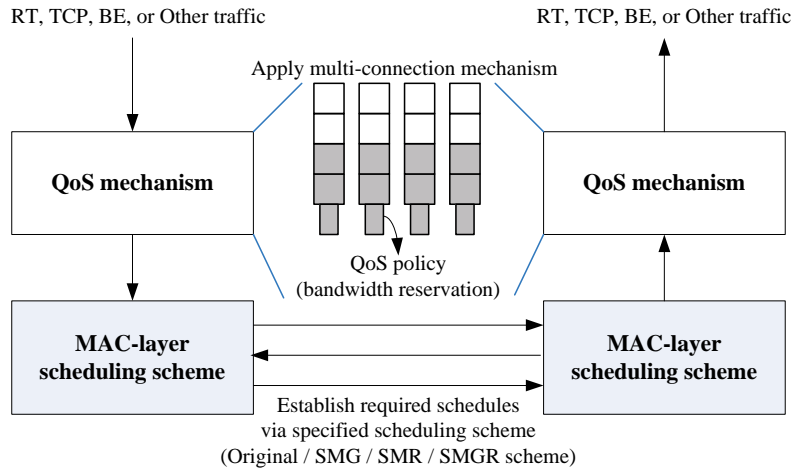


Figure 4.7: The system architecture when the QoS mechanism is applied

flows can obtain their required bandwidth in every three-way handshake procedure. Therefore, each traffic flow can be more smoothly served.

On the other hand, the SMG scheme has great effects on the soft-QoS support. This is because using the SMG scheme, the granting node can aggressively satisfies the requests, each of which specifies required bandwidth for each traffic flow. Thus, some QoS requirement such as bandwidth reservation can be better supported. Besides, since the SMG scheme can exploit the available fragmented network resources, these traffic flow can use these fragmented schedules to transmit their packets.

To fairly compare the soft-QoS support of the SMGR scheme against that of the original scheme, we also apply the multi-connection mechanism to the original scheme. In addition, we choose the “bandwidth reservation” as a performance metric to evaluate how well the SMGR and original schemes support the soft-QoS requirement. Thus, as shown in Fig. 4.7, we can compare the performance of soft-QoS support when using different MAC-layer scheduling schemes under the same “QoS mechanism.” The “QoS mechanism” consists of two functionalities, which are the multi-connection mechanism and the soft-QoS requirement in terms of bandwidth reservation, respectively.

Based on the same QoS mechanism, we evaluate the performances of soft-QoS support when using four different schemes. They are the original scheme, the SMR scheme, the SMG scheme and the SMGR scheme, respectively. The simulation results are shown in Section 6.2.

Chapter 5

Functionality Validation

In this chapter, we first compute the theoretical capacity at the MAC layer based on a permanent allocation policy and on a on-demand allocation policy. The derived throughputs are then compared with the application throughputs obtained from our simulation results. Next, we validate the implementation of the slicing-based scheduling scheme. Note that in this chapter, we do not verify the Mesh operations, such as the network entry process and distributed election-based scheduling, because they have been validated elaborately in [3].

Throughout this chapter, a simple one-link case is used to simplify the validation. This simulation case comprises two nodes, one of which runs a greedy UDP sender program while the other runs a corresponding UDP receiver program. The common simulation parameters are listed in Table 5.1. Note that the MSH-DSCH-NUM value is set to 8, which indicates that only the distributed scheduling is used.

5.1 Corresponding Throughputs

In this section, we validate the application throughputs using two kinds of bandwidth allocation policies, which are the permanent allocation policy and the on-demand allocation policy, respectively. The former is used to maximize MAC-layer throughput (i.e., the minislots can be fully utilized because of permanent allocation). The latter is used to obtain MAC-layer throughput, which all the schedules are established via

Table 5.1: The used simulation parameters in Chapter 5

Parameter Name	Value	Description
Xmt Holdoff Exponent	1	Xmt Holdoff Time = $2^{1+4} = 32$ TxOpps
MSH-CTRL-LEN	8	TxOpps per frame
MSH-DSCH-NUM	8	MSH-DSCH TxOpps per frame
Scheduling Frames	2	There are $(2 \cdot 4)$ schedule control subframes between two network control subframes
Frame Duration	10 ms	
Frame Frequency	1/Frame Duration	Frames per second
T_b	11.1 us	OFDM useful symbol time
T_g	$T_b/4$	OFDM guard time
UDP Payload Size	1472 bytes	Packet size except IP and UDP headers
PDU Size	1512 bytes	Size of non-fragmented PDU packet

a three-way handshake procedure. Thus, the wasted bandwidth contributed by the on-demand overheads must be considered.

5.1.1 Permanent allocation policy

The permanent allocation policy is used to validate the throughput result when whole network bandwidth are used by an application program. Based on this policy, a schedule between the sender and the receiver has been assigned in advance. To fully utilize the available bandwidth, this schedule is a permanent distributed allocation spanning whole data subframe.

In the following, we first compute the theoretical capacity at the MAC layer, and then the derived throughput is compared with the application throughput obtained from our simulation results. Besides, we discuss the overheads that result in the difference between the theoretical MAC-layer throughput and the realistic application throughput.

Let T_{Ctrl} and T_{Data} denote the throughput of the control subframe and of the data

subframe, respectively; S_{Frame} denotes the number of OFDM symbols per frame; S_{Ctrl} and S_{Data} denote the number of OFDM symbols in the control subframe and in the data subframe, respectively. Therefore, the MAC-layer throughput, T_{MAC} , can be expressed as:

$$T_{MAC} = T_{Ctrl} + T_{Data}$$

There are MSH-CTRL-LEN TxOpps per control subframe. Each TxOpp consists of 7 OFDM symbols. So we have

$$\begin{aligned} S_{Ctrl} &= MSH-CTRL-LEN \cdot 7 \\ &= 56 \end{aligned}$$

$$\begin{aligned} S_{Data} &= S_{Frame} - S_{Ctrl} \\ &= \frac{Frame\ Duration}{T_g + T_b} - S_{Ctrl} \\ &= 665 \end{aligned}$$

All transmissions in the control subframe are sent using the QPSK-1/2 mode. The size of a uncoded block in this mode is 24 bytes (in Table 3.2). Thus, T_{Ctrl} is a fixed value and can be computed as:

$$\begin{aligned} T_{Ctrl} &= \frac{S_{Ctrl} \cdot 48}{Frame\ Duration} \\ &= 134.4\ Kbyte/sec \\ &= 1.08\ Mbit/sec \end{aligned}$$

T_{Data} depends on the type of the PHY mode used. For example, when operating in the 64QAM-3/4 mode, with 108 bytes uncoded block size, T_{Data} can be computed as:

$$\begin{aligned} T_{Data} &= \frac{S_{Data} \cdot 108}{Frame\ Duration} \\ &= 7182\ Kbyte/sec \\ &= 57.46\ Mbit/sec \end{aligned}$$

Table 5.2: Theoretical MAC-layer throughput and realistic application throughput using mandatory PHY modes

	Theoretical MAC-layer Throughput (Mbit/sec)	Realistic Application Throughput (Mbit/sec)
BPSK-1/2	7.46	6.18
QPSK-1/2	13.84	12.38
QPSK-3/4	20.23	18.58
16QAM-1/2	26.61	24.77
16QAM-3/4	39.38	37.17
64QAM-2/3	52.15	49.56
64QAM-3/4	58.53	55.76

Consequently, the MAC-layer throughput, T_{MAC} , is $1.08 + 57.46 = 58.54 \text{ Mbit/sec}$.

The theoretical MAC-layer throughput using mandatory PHY modes is shown in the second column of Table 5.2. In addition, the realistic application throughput obtained from our simulation results is shown in the third column of Table 5.2, which is compared with the theoretical MAC-layer throughput.

The difference between the theoretical MAC-layer throughput and realistic application throughput is attributed to protocol overheads and MAC management messages exchanged in the control subframe. The protocol overheads consist of the UDP/IP headers, the MAC generic header/subheaders/tailer (CRC). We compute the overheads as follows.

Let O_{Ctrl} denote the overhead of MAC management messages exchanged in the control subframe; O_{Data} denotes the header overhead of transmit packets in the data subframe. Therefore, the MAC overhead, O_{MAC} , can be expressed as:

$$O_{MAC} = O_{Ctrl} + O_{Data}$$

Next, let H_{UDP} denote the UDP header size; H_{IP} denotes the IP header size; H_{PDU} denotes the PDU header size; H_{total} denotes the total header size per PDU packet. The overhead in the data subframe, O_{Data} , can be computed as:

$$O_{Data} = \frac{\text{number of transmit PDUs per frame} \cdot H_{total}}{\text{Frame Duration}}$$

The *max throughput per data subframe* is defined as $S_{Data} \cdot \text{uncoded block size in the}$

Table 5.3: The parameters used in on-demand allocation policy

Parameter Name	Value	Description
REQ-SLOT-SIZE	30	Demand in minislots
REQ-FRAME-SIZE	32	Persistent frames for demand minislots
AVAIL-SLOT-SIZE	50	The number of minislots free for grants
AVAIL-FRAME-SIZE	32	Persistent frames for the free minislots

data subframe (i.e., is 71820 bytes). Besides, as shown in Table 5.1, the PDU size is 1512 bytes. Thus,

$$\begin{aligned} \text{number of transmit PDUs per frame} &= \text{ceil}\left(\frac{\text{max throughput per data subframe}}{\text{PDUSize}}\right) \\ &= 48 \end{aligned}$$

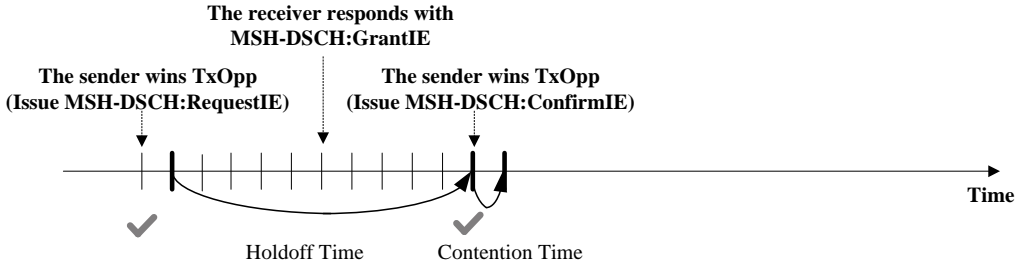
Here, we assume that the UDP packets are not fragmented in the MAC layer (i.e., only one UDP packet is encapsulated into one PDU). Therefore, the H_{total} can be computed as:

$$\begin{aligned} H_{total} &= H_{PDU} + H_{IP} + H_{UDP} \\ &= 12 + 20 + 8 \text{ (bytes)} \\ &= 40 \text{ (bytes)} \end{aligned}$$

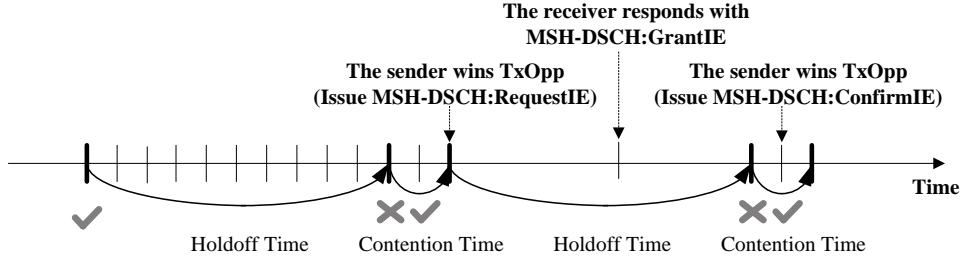
Finally, the overhead in the data subframe, O_{Data} , is $48 \cdot 40 \text{ bytes} / 10 \text{ ms} = 1.54 \text{ Mbit/sec}$. In addition, the overhead in the control subframe, O_{Ctrl} is equal to T_{Ctrl} . Consequently, the MAC overhead, O_{MAC} , is $1.08 + 1.54 = 2.62 \text{ Mbit/sec}$.

5.1.2 On-demand allocation policy

In this policy, a schedule between the sender and the receiver has to be established in an on-demand basis. This is accomplished by a three-way handshake procedure. In the following, we first discuss the theoretical minislot utilization by an analytical method, and then figure out the MAC-layer throughput based on this obtained



(a) The required TxOpps in the best case



(b) The required TxOpps in the worst case

Figure 5.1: The required TxOpps in the schedule on-demand process (a) in the best case and (b) in the worst case

minislot utilization. Furthermore, the derived throughput is then compared with the throughput obtained from our simulation results. The simulation parameters of this on-demand allocation policy are listed Table 5.3.

Let F_{Wasted} denote the number of wasted frames contributed by the on-demand allocation policy, which indicates that there are no data transmitted in this period; F_{Unused} denotes the number of frames unused to transmit any data per second; F_{Used} denotes the number of frames used to transmit data per second; $M_{Granting}$ denotes the number of a node's granting minislots per frame. It can be expressed as:

$$M_{Granting} = \frac{REQ-SLOT-SIZE \cdot F_{Used}}{Frame Frequency} \quad (5.1)$$

Next, let $T_{Granting}$ denote the derived MAC-layer throughput when fully utilizing the granting minislots. Each minislot consists of 3 OFDM symbols, and the uncoded block size is 108 bytes when the 64QAM-3/4 mode is used. So we have

$$T_{Granting} = \frac{M_{Granting} \cdot 3 \cdot 108}{Frame Duration} \quad (5.2)$$

The wasted frames contributed by the on-demand allocation policy, F_{Wasted} , greatly influences the delays experienced by the sender and receiver. Furthermore, it may have further impacts on the derived MAC-layer throughput. As a result, we discuss the possible value of F_{Wasted} , and then the MAC-layer throughput is figured out based on this derived F_{Wasted} . In the following, both of the best case and the worst case are considered.

Best Case:

In such a case, we make two assumptions: (1) The sender always can win the first TxOpp during its contention time, and (2) in the schedule on-demand process, only the schedule control subframes are involved (i.e., except the network control subframe). In the following, we describe this process in more detail.

As shown in Fig. 5.1 (a), the sender issues the MSH-DSCH:RequestIE upon winning the first TxOpp. During its holdoff time, the receiver responds with the MSH-DSCH:GrantIE specifying the assigned schedule. After waiting the holdoff time, the sender starts to contend for the transmission opportunity to confirm its schedule. Based on the above assumption, the sender soon wins the first TxOpp, sending back the MSH-DSCH:ConfirmIE to the receiver. Therefore, the F_{Wasted} can be computed as:

$$\begin{aligned} F_{Wasted} &= \text{ceil}\left(\frac{1 + \text{Holdoff Time} + 1}{\text{MSH-CTRL-LEN}}\right) \\ &= 4 \end{aligned}$$

In the best case, we should consider the slightest overhead of wasted frames to obtain the best MAC-layer throughput. As shown in Fig. 5.2 (a), the F_{Used} should be computed as $\text{Frame Frequency} - 2 \cdot F_{Wasted}$. Thus, based on Eq. (5.1), the $M_{Granting}$ can be computed as:

$$\begin{aligned} M_{Granting} &= \frac{30 \cdot (\text{Frame Frequency} - 2 \cdot F_{Wasted})}{100} \\ &= 27.6 \end{aligned}$$

Eventually, based on the Eq. (5.2), we compute the theoretical MAC-layer through-

put, $T_{Granting}$, as $27.6 \cdot 3 \cdot 108 \text{ bytes}/10 \text{ ms} = 7.15 \text{ Mbit}/\text{sec}$.

Worst Case:

In such a case, we also make two assumptions: (1) The sender always wins the last TxOpp during its contention time (i.e., it wins the competing TxOpp until no one is eligible to contend for the transmission opportunity), and (2) in the schedule on-demand process, extra two network control subframes must be considered. In the following, we describe this process in more details.

As shown in Fig. 5.1 (b), the sender first is refrained from contending for the transmission opportunities during its holdoff time because the sender has used the last TxOpp to exchange its MSH-DSCH message. During its holdoff time, the receiver responds with the MSH-DSCH:GrantIE specifying the assigned schedule. After waiting the holdoff time, the sender starts to compete for the transmission opportunity. Based on the above assumption, it wins the last TxOpp to transmit its MSH-DSCH:RequestIE. Similarly, the further holdoff time and contention time are necessary for replying its confirm message. Consequently, the F_{Wasted} can be computed as:

$$\begin{aligned}
 F_{Wasted} &= \text{ceil}\left(\frac{2 \cdot (\text{Holdoff Time} + 2) + 2 \cdot \text{MSH-CTRL-LEN}}{\text{MSH-CTRL-LEN}}\right) \\
 &= 10
 \end{aligned}$$

In the worst case, the heaviest overhead of wasted frames should be considered to acquire the worst MAC-layer throughput. As shown in Fig. 5.2 (b), the F_{Used} should be computed as $\text{Frame Frequency} - 3 \cdot F_{Wasted}$. Thus, based on Eq. (5.1), the $M_{Granting}$ can be computed as:

$$\begin{aligned}
 M_{Granting} &= \frac{30 \cdot (\text{Frame Frequency} - 3 \cdot F_{Wasted})}{100} \\
 &= 21
 \end{aligned}$$

Finally, based on the Eq. (5.2), the theoretical MAC-layer throughput, $T_{Granting}$, is computed as $21 \cdot 3 \cdot 108 \text{ bytes}/10 \text{ ms} = 5.44 \text{ Mbit}/\text{sec}$.

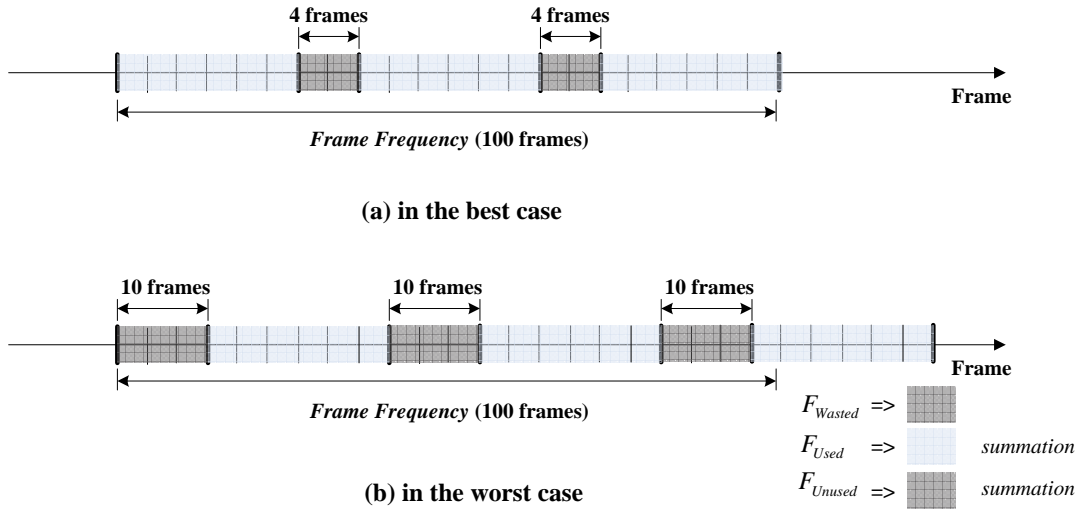


Figure 5.2: The overhead of wasted frames (a) in the best case and (b) in the worst case

After above analyses, we compare the derived $M_{Granting}$ and MAC-layer throughput with those obtained from our simulation results. The obtained realistic $M_{Granting}$ is 24.0, and realistic application throughput is 5.86 *Mbit/sec*. One sees that the values of them are between those in the worst case and those in the best case.

5.2 Slicing-based Scheduling Scheme

As mentioned in Section 4.2, the slicing-based scheduling scheme supplies two mechanisms: the multi-grant mechanism and the multi-request mechanism. Thus, it can support the SMG scheme when applying the multi-grant mechanism, can support the SMR scheme when applying the multi-request mechanism. In the following, we validate both the implementations of the SMG scheme and the SMR scheme by conducting a series of simulations. The simulation parameters used are presented in Table 5.1 and Table 5.3.

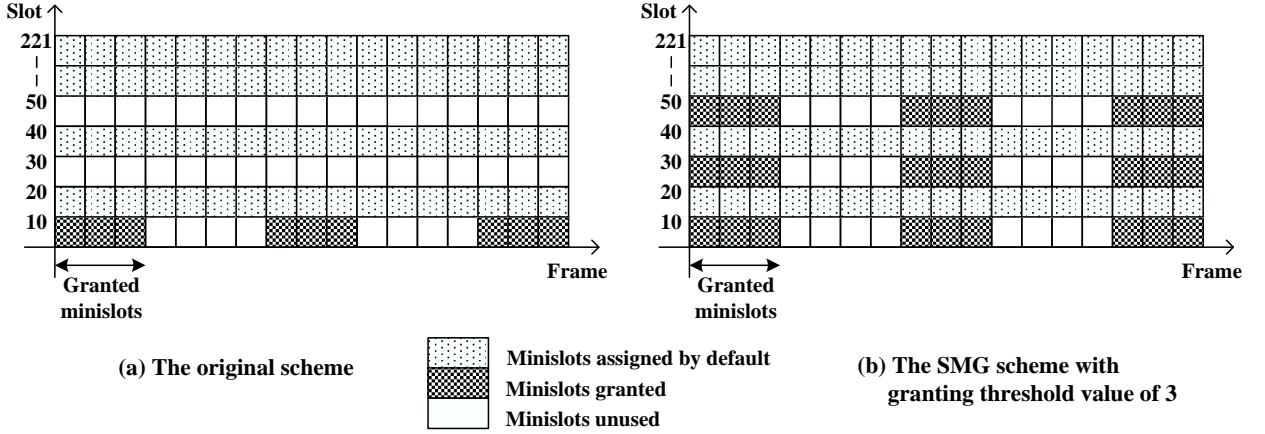


Figure 5.3: The difference of granting resources between using (a) the original scheme and (b) the SMG scheme with granting threshold value of 3

5.2.1 SMG Scheme

As mentioned in Section 4.2.1, the SMG scheme is proposed to improve the network bandwidth utilization. In the following, we set up a simulation case to examine our implementation of the SMG scheme.

To fully present the advantages of the SMG scheme, some schedules are assigned by default, as shown in Fig. 5.3. Therefore, in such a network, only partial bandwidth can be scheduled.

We choose both of a node's granting minislots per frame, $M_{Granting}$, and application throughput, $T_{Granting}$, to be our performance metrics. Intuitively, the results of these metrics should be proportion to the granting threshold value. Fig. 5.3 is an example that shows the minislot utilization difference between using the original scheme and using the SMG scheme with G_{Thres} value of 3.

Table 5.4 shows the $M_{Granting}$ and $T_{Granting}$ results, which are obtained when using the original scheme and the SMG scheme with different granting threshold value. One sees that there are two significant results: (1) as the granting threshold value increases, the $M_{Granting}$ and $T_{Granting}$ also increase and (2) the improved ratio of the $M_{Granting}$ is equal to that of $T_{Granting}$, which indicates the improvement of

Table 5.4: A node’s granting minislots per frame and application throughput of the SMG scheme with different granting threshold value and the original scheme

	A Node’s Granting Minislots Per Frame ($M_{Granting}$)			Application Throughput (Mbit/sec) ($T_{Granting}$)		
	Original	SMG	Improved Ratio	Original	SMG	Improved Ratio
G_{Thres} 1	8.00	8.00	1.0	1.95	1.95	1.0
G_{Thres} 2	8.00	16.00	2.0	1.95	3.91	2.0
G_{Thres} 3	8.00	24.00	3.0	1.95	5.87	3.0

application performance is attributed to the more efficient minislot utilization.

5.2.2 SMR Scheme

As mentioned in Section 4.2.2, the SMR scheme is designed to more smoothly serve the end-to-end traffic flows. In the following, we conduct a simulation case to validate our implementation of the SMR scheme.

To simplify the validation of the SMR scheme, we run three greedy UDP traffic flows on the sender, and then these flows should be classified into the corresponding queues, as shown in Fig. 5.4. There are two reasons for this: (1) eliminate the effects of upper layer behavior (such as TCP) , and (2) keep these queues in the MAC layer always busy to fully utilize the scheduled resources. Besides, for the original scheme, we adopt the round-robin policy that serve each traffic flow in turn.

Both of a node’s requesting minislots per frame and application throughput are chosen as the performance metrics used to valid the SMR scheme. Intuitively, the results of theses metrics using the SMR scheme should be proportion to those using the original scheme in the whole view. Take Fig. 5.4 as an example, we see that the there is difference between the original scheme and the SMR scheme when the sender issues its requests.

Table 5.5 shows the simulation results of requested minislots per frame and application throughput for each traffic flow, which uses the original scheme and the SMR scheme individually. Two important results are obtained. First, for these performance metrics, on average our SMR scheme speeds up the original scheme by 3 times. This is because the SMR scheme can simultaneously issue 3 requests,

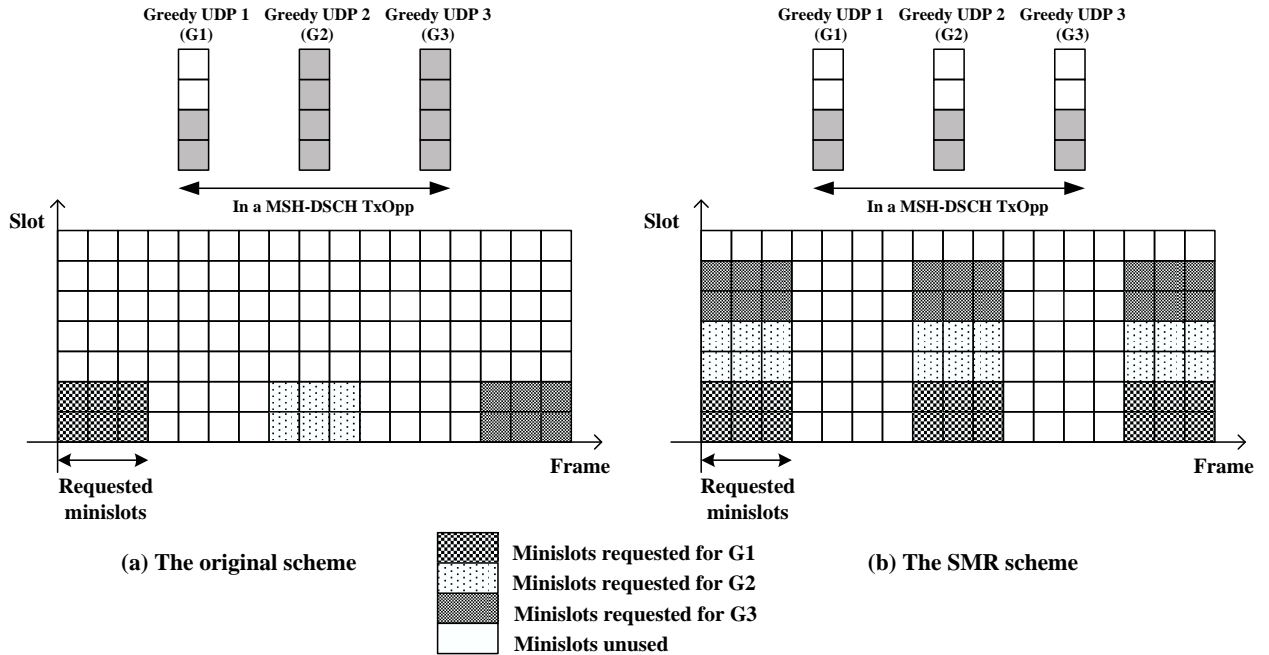


Figure 5.4: The difference of requesting resources between using (a) the original scheme and (b) the SMR scheme

which every request serves individual traffic flow, in a MSH-DSCH TxOpp. Second, for the original scheme, the number of requested minislots per frame for each traffic flow are unstable. Although the round-robin policy is adopted, the traffic flow still may not be served during its turn. This is because this traffic flow will lose its turn that there has been a schedule established between the communicating peers.

5.3 The Soft-QoS Support

As mentioned in Section 4.3, the bandwidth reservation mechanism is used to be a performance metric, which evaluates the support of the soft-QoS policy when the SMR scheme is used or not. In this section, we examine our implementation of this mechanism by simulations. The simulation parameters used are listed in Table 5.1.

We run four different traffic flows between the sender and receiver, which these flows are TCP, greedy UDP, CBR with 100 Kbyte/sec, and CBR with 10 Kbyte/sec,

Table 5.5: A node’s requesting minislots per frame and application throughput using the original scheme and the SMR scheme

	A node’s requesting minislots Per Frame			Application Throughput (Mbit/sec) ($T_{Granting}$)		
	Original	SMR	Ratio	Original	SMR	Ratio
1st Greedy UDP Flow	16.32	24.00	1.5	3.93	5.88	1.5
2nd Greedy UDP Flow	2.88	24.00	8.3	0.81	5.88	7.3
3rd Greedy UDP Flow	4.80	24.00	5.0	1.10	5.88	5.3
Total Greedy UDP Flows	24.00	72.00	3.0	5.84	17.64	3.0

Table 5.6: The requested bandwidth reservation and the obtained application throughput for different traffic flows

	Requested Bandwidth Reservation (Kbyte/sec)	Obtained Application Throughput (Kbyte/sec)	Achievement Ratio (%)
TCP Flow	500.00	498.20	99.64
Greedy UDP Flow	300.00	299.89	99.96
CBR with 100 Kbyte/sec	80.00	79.98	99.98
CBR with 10 Kbyte/sec	5.00	4.98	99.60

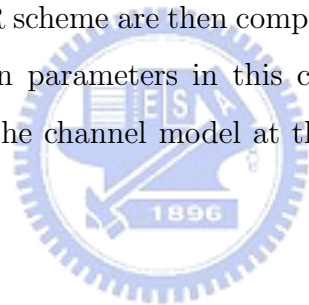
respectively. Then the obtained application throughputs will be compared with our bandwidth reservation setting. To simplify the validation of this bandwidth reservation mechanism, we establish four permanent distributed schedules for each traffic flow in advance (i.e., the minislots are divided into four equal parts to respectively supply the data transmissions to these flows). In addition, the additional permanent schedule is also set up for TCP ack transmission, which uses the remaining minislots established previously.

Table 5.6 shows the obtained application throughputs using our bandwidth reservation setting. As expected, the obtained throughputs are almost equal to our setting values for each traffic flow.

Chapter 6

Performance Evaluation

In this chapter, we first evaluate the performances of the SMG scheme and the original scheme. The soft-QoS support of the SMBG scheme, that of the SMR scheme and that of the SMGR scheme are then compared against that of the original scheme. The used simulation parameters in this chapter are listed in Table 6.1. Besides, in our simulations, the channel model at the physical layer is disabled by default.



6.1 Simulation Results on The SMG Scheme

When the distributed scheduling is used, the three-way handshake procedure used for establishing data schedules may incur an unavoidable large delay. As mentioned in Section 4.2.1, we propose the SMG scheme to aggressively satisfy the required bandwidth between peer nodes. Thus, the required time for establishing a data schedule can be reduced. Using this scheme, the network bandwidth is utilized more efficiently, and thus the application performances can be significantly improved.

In this section, we first evaluate the proposed SMG scheme under three different networks and then discuss the network scalability issue. Next, the effects of an important parameter “availability slot size” on the proposed SMG scheme and the original scheme are also presented.

Table 6.1: The used simulation parameters in Chapter 6

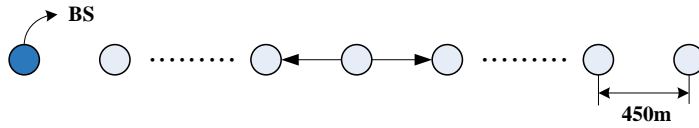
Parameter Name	Value	Description
Xmt Holdoff Exponent	1	Xmt Holdoff Time = $2^{1+4} = 32$ TxOpps
MSH-CTRL-LEN	8	TxOpps per frame
MSH-DSCH-NUM	8	MSH-DSCH TxOpps per frame
Scheduling Frames	2	There are $(2 \cdot 4)$ schedule control subframes between two network control subframes
REQ-SLOT-SIZE	30	Demand in minislots
REQ-FRAME-SIZE	32	Persistent frames for demand minislots
AVAIL-SLOT-SIZE	50	The number of minislots free for grants
AVAIL-FRAME-SIZE	32	Persistent frames for the free minislots
PHY mode	64QAM-3/4	108 bytes per uncoded block
Frame Duration	10 ms	

6.1.1 Used Network Configuration

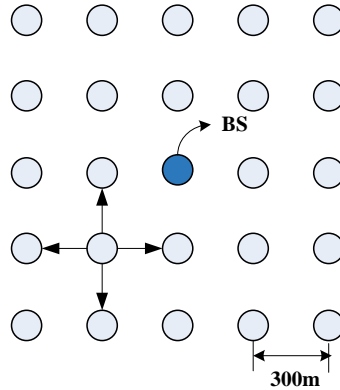
We use three different network topologies to evaluate the performances of the SMG scheme and the original scheme. They are chain, grid and random network topologies, respectively. In the following, we present the three network topologies, respectively. In addition, the used network configurations and traffic patterns for each topology are also described.

As shown in Fig. 6.1 (a), the chain network comprises 21 nodes. The left-most one is the BS node, and the others are SS nodes. Each of these node is spaced 450 meters apart from its left and right neighbors. With such an arrangement, a node can only transmit its data to its two neighboring nodes because the transmission range of each node is set to 500 meters. Besides, in this chain network, each node is configured to establish connections with its two neighbors.

As shown Fig. 6.1 (b), we create a 5x5 grid network comprising 25 nodes, each of which is spaced 300 meters apart from its vertical and horizontal neighbors. The BS node is located in the center of this grid network. Similarly, in this grid network,



(a) a chain topology consisting of 21 nodes



(b) a 5x5 grid topology

Figure 6.1: (a) a chain network consisting of 21 nodes and (b) a 5x5 grid network used in Section 6.1

each node is configured to establish connections with its vertical and horizontal neighbors.

For random network simulations, we randomly create a network consisting of 100 nodes. The BS and SS nodes are randomly distributed in this network. In this network, 100 connections are randomly established between 100 pairs of nodes.

6.1.2 Minislot Utilization and Application Throughput

In this section, we evaluate the performances of the SMG scheme with different granting threshold indices and the original scheme under three different network topologies. They are chain, 5x5 grid, and random network topologies respectively, as described in Section 6.1.1. Besides, we run these schemes on each topology 3 times, each time using a different random number seed. The simulated time of each run is set to 400 seconds.

For the chain network, we perform two suites of performance tests to observe the performances of UDP/TCP traffic over such a network. In the first suite, each node

generates greedy UDP traffic to its two neighboring nodes. In the second suite, each node establishes TCP connections with its two neighbors instead. In contrast, only greedy UDP traffic is generated in the 5x5 grid and random networks. In the former case, like the chain network scenario, each node generates greedy UDP traffic to its neighbors. In the latter case, 100 UDP flows are randomly generated between 100 pairs of distinct nodes in this network (i.e., a network node is either a source node or a destination node of a flow and does not belong to two flows simultaneously.). Thus, several flows may get across multiple hops.

Four performance metrics are used to evaluate the performance gain of the SMG scheme. The first one is $M_{Granting}$, indicating the number of a node's granting minislots per frame; the second one is M_{Util} , indicating the total use of data minislots within a node's two-hop neighborhood. As mentioned in Section 5.1.2, these two performance metrics are used to quantify the MAC-layer performances regarding scheduling efficiency and bandwidth utilization. The last two performance metrics are average throughput and average packet delay, which are used to evaluate the Application-layer performances. The average throughput is defined as the average of throughputs obtained by all flows. The average packet delay is defined as the average of the required time for packets to travel from their senders to their receivers.

To study the effect of the granting threshold parameter, we use three different values for this parameter to compare the resultant performances of our proposed SMG scheme. The “ G_{Thres} maximum” denotes a scheme that the granting threshold value in the SMG scheme is set to the maximum value defined in the standard; the “ G_{Thres} 3” indicates a scheme that the granting threshold value in the SMG scheme is set to 3; and the “ G_{Thres} 6” indicates a scheme that the granting threshold value of the SMG scheme is set to 6. Recall that using the original scheme, only one data schedule can be assigned to a request during a three-way handshake procedure. Thus, the original scheme is equivalent to the SAMG scheme with G_{Thres} value being set to 1. It is denoted by “Original.”

As shown in Table 6.2 (a), in the chain network, as the value of the granting threshold increases, the $M_{Granting}$ value and M_{Util} value increase. These results

show that when the granting node can assign more data schedules for the incoming requests, both the average of a node's granting minislots per frame and the average of minislot utilization increase. Besides, application throughputs are also significantly improved because the minislots are more efficiently utilized when our proposed scheme is used. Our SMG scheme outperforms the original scheme by a factor of 1.419 on $M_{Granting}$ regarding UDP traffic and by a factor of 1.540 on $M_{Granting}$ regarding TCP traffic, respectively.

As we can see Table 6.2(b), in the chain network, under UDP traffic $M_{Granting}$ can be increased by a factor of 1.419 while under TCP traffic $M_{Granting}$ can be increased by a factor of 1.540. These increases can be directly reflected in the data minislot utilization. Under UDP traffic, the data minislot utilization can be increased by a factor of 1.415 while under TCP traffic the data minislot utilization can be increased by a factor of 1.537.

In the following, we explain why TCP traffic can be served much better under the SMG scheme. TCP dynamically adjusts its transmission rate based on network conditions and thus the required MAC-layer bandwidth varies over time. In such a condition, the requested size issued by the MAC layer differs over time, which can lead to drastic fragmentation regarding available network bandwidth. This is because in an IEEE 802.16 Mesh network, each bandwidth request should be assigned a consecutive data minislots and therefore variable-sized requests can result in data schedules that cannot be aligned. As such, as compared to the original scheme, the proposed SMG scheme can more efficiently utilize the fragmented network bandwidth and achieve better MAC-layer performances.

Regarding application performances, in this chain network, the SMG scheme outperforms the original scheme on UDP throughputs by a factor of 1.42 and on TCP throughputs by a factor of 1.32, respectively. The discrepancy between the improvement ratios of application throughputs and that of data minislot utilization is due to different behaviors of application programs. As for greedy UDP traffic, it sends data packets as much as possible and thus it can fully utilize the available MAC-layer bandwidth. In contrast, TCP employs flow control and congestion con-

Table 6.2: The performances of the SMG scheme with different granting threshold indices and the original scheme

(a) Chain network comprising 21 nodes runs UDP connections

	UDP connection			
	MAC		Application	
	$M_{Granting}$	M_{Util}	Throughput (Kbyte/sec)	Packet Delay Time (ms)
G_{Thres} maximum	50.75	67.25%	833.20	90.97
G_{Thres} 6	48.18	63.84%	790.91	96.03
G_{Thres} 3	45.58	60.34%	748.17	102.04
Original	35.77	47.52%	586.81	128.92

(b) Chain network comprising 21 nodes runs TCP connections

	TCP connection		
	MAC		Application
	$M_{Granting}$	M_{Util}	Throughput (Kbyte/sec)
G_{Thres} maximum	41.27	54.74%	299.43
G_{Thres} 6	39.56	52.45%	290.60
G_{Thres} 3	37.99	50.32%	275.74
Original	26.80	35.60%	226.20

(c) 5x5 grid network

	UDP connection		
	MAC		Application
	$M_{Granting}$	M_{Util}	Throughput (KB/sec)
G_{Thres} maximum	23.56	71.61%	296.16
G_{Thres} 6	23.47	71.26%	298.33
G_{Thres} 3	23.28	70.59%	296.64
Original	18.31	57.96%	202.05

(d) Random network comprising 100 nodes

	UDP connection		
	MAC		Application
	$M_{Granting}$	M_{Util}	Throughput (KB/sec)
G_{Thres} maximum	21.46	69.03%	161.72
G_{Thres} 6	20.61	66.57%	158.85
G_{Thres} 3	19.60	63.67%	152.36
Original	15.41	52.78%	107.93

trol mechanisms to throttle its transmission rate and thus cannot fully utilize the underlying available MAC-layer bandwidth.

Besides, the SMG scheme can on average reduce the packet delay by a factor of 1.42, when compared with the original scheme. This is because the SMG scheme can assign more than one schedule for a request during a three-way handshake procedure and thus reduce the required time for allocating multiple data schedules, as compared to the original scheme. As such, the delay experienced by data packets can be greatly decreased.

As shown in Table 6.2 (a) and (b), in the 5x5 grid and random networks, the $M_{Granting}$ result shows that the SMG scheme can on average outperform the original scheme in the number of a node's granting minislots per frame. We can also see that as the G_{Thres} value increases, the number of a node's granting minislots per frame increases because larger G_{Thres} values can provide more flexibility for the smg scheme to grant more minislots.

The M_{Util} result shows that the SMG scheme is superior to the original scheme in bandwidth utilization of each node's two-hop neighborhood. Similar to the $M_{granting}$ result, as the G_{thres} value increases, the bandwidth utilization also increases. The reason has been explained above.

Besides, recall that $M_{granting}$ denotes the number of a node's granting minislots per frame (per 221 minislots), while M_{util} denotes the total use of data minislots within a node's two-hop neighborhood. As can be seen in Table 6.2 (b), (c) and (d), although the SMG scheme achieves the lowest bandwidth utilization in the chain network case, nodes in the chain network case can grant the most number of data minislots. The discrepancy between the $M_{granting}$ and M_{util} results is because the average number of contending nodes in a node's two-hop neighborhood is much less in the chain network case than in other network cases. As such, although the SMG scheme can achieve higher utilization in the grid and random network cases, the increased network bandwidth is shared by a large number of contending nodes. As a result, the average number of data minislots that a node can grant in these two network cases is much less than that in the chain network case.

6.1.3 Network Scalability Issue

In this section, we evaluate the MAC-layer bandwidth utilization and total network capacity under different network scales using the four schemes mentioned above. The used network topologies are the 3x3, 4x4, 5x5 and 7x7 grid network topologies, which the 5x5 grid network have been described in Section 6.1.1. The simulation scenarios of these other network topologies are similar to that of the 5x5 network topology, in which each node generates greedy UDP traffic to all of its vertical and horizontal neighbors. Besides, we run these schemes on each topology 3 times, each time using a different random number seed. The simulated time of each run is 400 seconds.

As shown in Fig. 6.2, as the network scale enlarges, the minislot utilization increases. Recall that the minislot utilization indicates the total use of data minislots within a node's two-hop neighborhood. In the larger network scales, a node should has more neighboring nodes in its two-hop neighborhood. These neighboring nodes will share the network bandwidth with this node. Thus, the minislot utilization can be more efficiently utilized in the large network scales than in the small network scales.

One can see that the SMG scheme is superior to the original scheme in the network bandwidth utilization under any network scale. This is because this SMG scheme can more aggressively exploit the available network bandwidth that may never be used in the original scheme. As a result, in any network scale the SMG scheme can more effectively utilize the network bandwidth than the original scheme.

Besides, the results show that as the value of the granting threshold increases, the network bandwidth utilization increases, especially for larger network scale (e.g., the 7x7 grid network). The reason is explained below. There are more contending nodes in the larger network scale than in the smaller network scale. These contending nodes share the network bandwidth and establish all kinds of data schedules according to their bandwidth requirement. These various data schedules are disorderly scattered within the network, and thus many fragmented available bandwidth

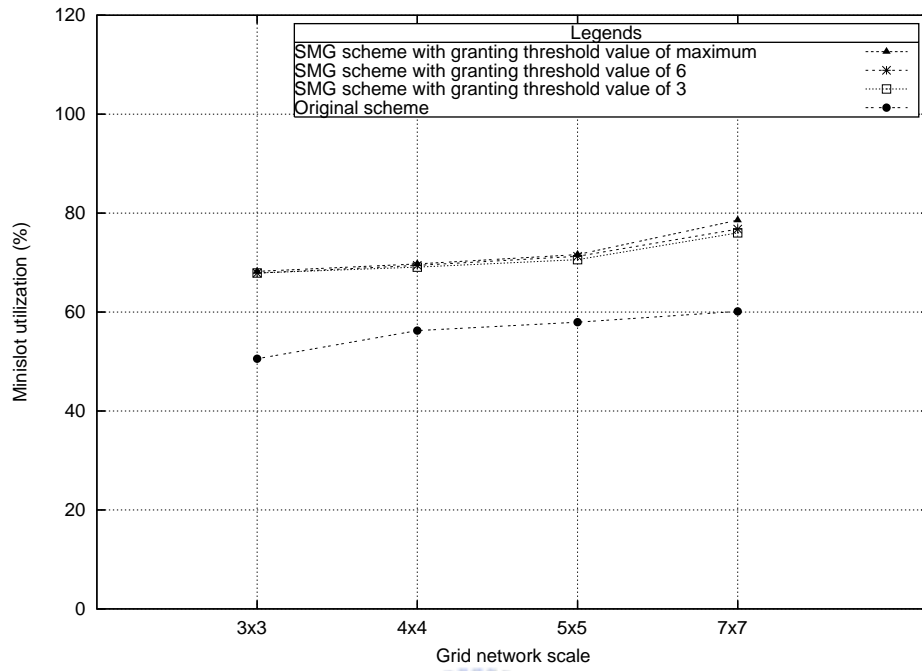


Figure 6.2: Minislot utilization vs. grid network scale

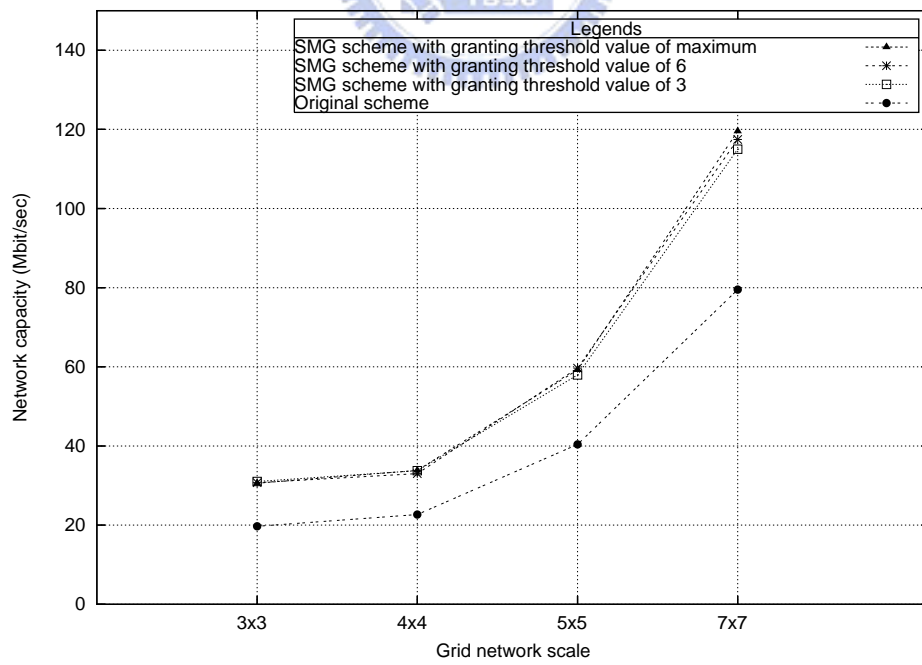


Figure 6.3: Network capacity vs. grid network scale

will form. As mentioned before, when the value of granting threshold is increased, the granting node can assign more data schedules for a request during a three-way handshake procedure. In such a condition, the fragmented available bandwidth can be efficiently exploited, and thus the network bandwidth utilization is enhanced.

The network capacity indicates that the total throughputs obtained by all traffic flows in the network. As mentioned before, the MAC-layer minislot utilization can be directly reflected in application performances. Thus, we can predict that the network capacity can be increased by the SMG scheme because this scheme can enhance the MAC-layer minislot utilization. Fig. 6.3 shows that the SMG scheme outperforms the original scheme by a factor 49% under the 4x4 grid network, by 47% under the 5x5 grid network and by 50% under the 7x7 grid network. The reasons for the network capacity improvement are similar to those for the minislot utilization improvement, which have been explained before.

6.1.4 Availability Slot Size Effects

As mentioned in Section 3.1.5, when the requesting node establishes a schedule with its neighbor, it issues a request IE and an availability IE. The former is used to specify the required bandwidth. The latter is used to indicate the free bandwidth range, in which the granting node can assign the schedules. The **Minislot range** is carried in the availability IE and is used to specify the free minislot range for granting, which is denoted by the “availability slot size” below.

Setting the value of the availability slot size for each request is essential to achieve good performances of the original and SMG schemes. In this section, we evaluate the performances of the original and SMG schemes when different availability slot sizes are used. The number of a node’s granting minislots per frame, $M_{Granting}$, and application throughput are selected as our performance metrics. The used network configuration is shown in Fig. 6.1 (a). We run the original and SMG schemes using different availability slot sizes 3 times, each time using a different random number seed. The simulated time of each run is 200 seconds.

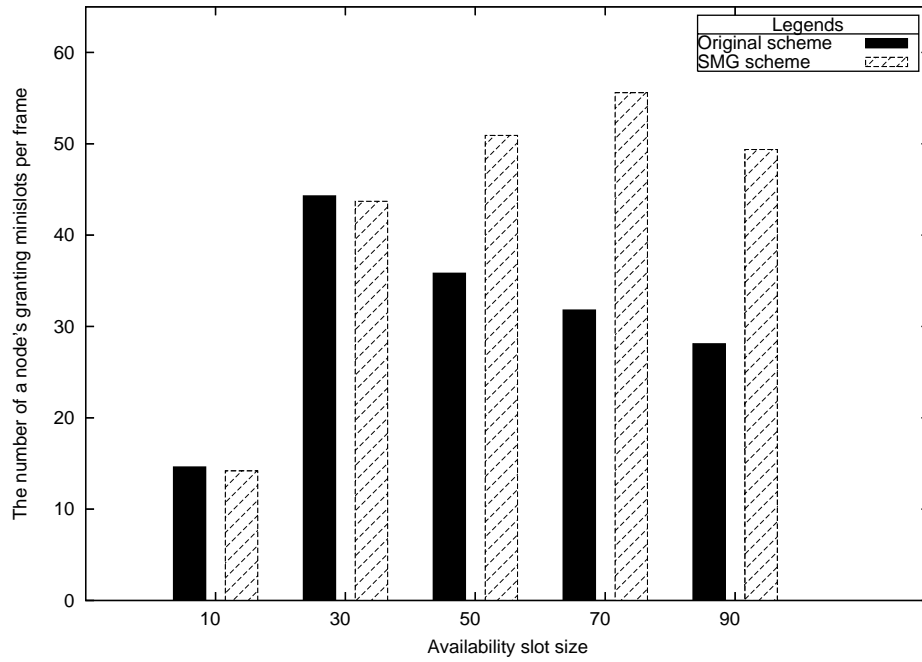


Figure 6.4: The effects of availability slot size on the number of a node's granting minislots per frame

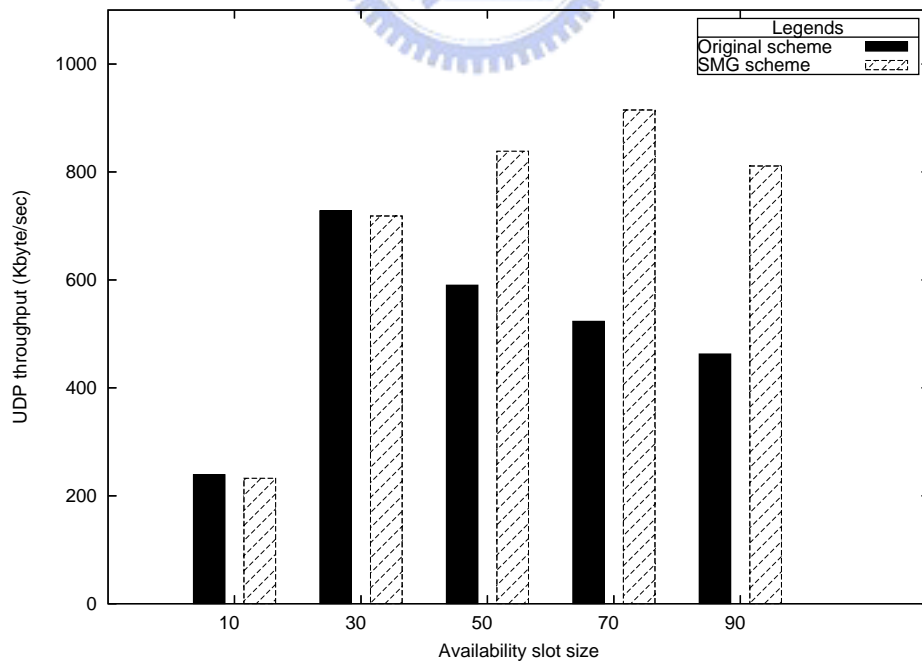


Figure 6.5: The effects of availability slot size on application performance

Simulation results are shown in Fig. 6.4 and Fig. 6.5. As shown in Fig. 6.4, when the availability slot size of 10 and 30 are used, the $M_{Granting}$ value of the SMG scheme is almost the same as that of the original scheme. This is because the value of the availability slot size is too small so that few available resources in the network can be exploited by our proposed scheme. Therefore, when the availability slot size of 50, 70 and 90 are used, the SMG scheme on average can improve the $M_{Granting}$ by 42%, 75% and 76%, respectively.

One sees that, however, the performance of our proposed scheme when the availability slot size of 90 is used worse than that when the availability slot size of 70 is used. The reason is that when the very large availability slot size is used, the nodes may establish its schedules very greedily, even overuse the available resources that may be assigned to their neighboring nodes originally. Therefore, the network performance declines due to such a unfairness of resource sharing.

When the value of the availability slot size exceeds 30, the $M_{Granting}$ of the original scheme goes down as this value increases. This is also attributed to the unfairness of resource sharing mentioned above. Furthermore, unlike the SMG scheme, the original scheme cannot utilize those fragmented resources, and thus the network performance becomes more and more inefficient as this value increases.

Fig. 6.5 shows the average UDP throughput when the different availability slot sizes are used. As mentioned before, we see that the resources utilization is more efficient, and better application throughput can be achieved.

6.2 Simulation Results on Soft-QoS Support

6.2.1 Used Network Configuration

In this section, we use a chain network topology to evaluate the performance of soft-QoS support. As shown in Fig. 6.6, this chain network consists of 21 nodes, each of which is spaced 450 meters apart from its left and right neighbors. From left to right, the nodes are named BS, SS(1), SS(2), ..., and SS(20), respectively. We

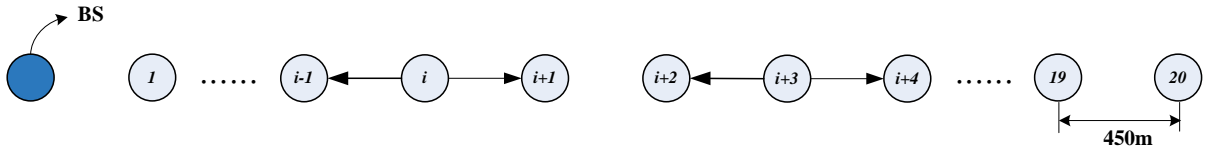


Figure 6.6: The network configuration used in Section 6.2

choose that the source nodes of traffic flows are at SS(1), SS(4), ..., SS(i), SS($i+3$), ..., SS(19) while the destination nodes of traffic flows are chosen to be these source nodes' left and right neighbors. These traffic flows are TCP, greedy UDP, CBR with 100 Kbyte/sec and CBR with 10 Kbyte/sec, respectively, which are set up between the source and destination nodes.

6.2.2 End-to-End Traffic

In this section, we first evaluate the MAC-layer performances when using four schemes, and then evaluate the application-layer performances of different end-to-end traffic flows when using four schemes. These four schemes are the “Original,” “SMG,” “SMR” and “SMGR,” respectively. The “Original” denotes the scheme that uses the original scheme when requesting and granting resources; the “SMG” denotes the scheme that uses the SMG scheme when granting resources and uses the original scheme when requesting resources; the “SMR” means the scheme that uses the SMR scheme when requesting resources and uses the original scheme when granting resources; the “SMGR” means using the SMG scheme when granting resources and using the SMR scheme when requesting resources.

The used network configuration is shown in Fig. 6.6. We run these four schemes based on this configuration 3 times, each time using a different random number seed. The simulated time of each run is 200 seconds.

As shown in Table 6.3 (a), the “SMGR,” “SMR” and “SMG” outperform the “Original” by a factor of 2.88, 1.94 and 1.74, respectively. The reasons why these schemes are superior to the original scheme have been explained in Section 4.3. Besides, ones see that the MAC-layer performance of the “SMR” is better than that

of the “SMG.” The reason is explained below. Although the “SMR” can only assign a schedule for a specific request, it can simultaneously issue at most four schedules, each of which is for a specific traffic flow, during a three-way handshake procedure. In contrast, although the “SMG” can assign more than one schedules for a specific request, it can only issue a request for a specific traffic flow during a three-way handshake procedure. As a result, the bandwidth utilization of the “SMR” is better than that of the “SMG”.

Table 6.3 (b) and (c) show the average application throughput and packet delay for four kinds of end-to-end traffic flows, each of which is set up between two peer nodes. Since the trends of the packet delay are similar to those of the application throughput, we only explain the results in terms of application throughput.

Regarding the application performances, we can see that the SMGR scheme outperforms the original scheme by a factor of 1.26 on TCP throughput, by a factor of 4.32 on UDP throughput, by a factor of 4.49 on CBR with 100 Kbyte/sec throughput, and by a factor of 1.95 on CBR with 10 Kbyte/sec throughput, respectively. Besides, one sees that the gaps of TCP performances among the four schemes are very small. In the following, we explain this phenomenon from two aspects.

First, in the “SMGR” and “SMR,” which the SMR scheme is used, we specify that the TCP traffic flow has the highest priority when the requesting node issues the requests for all traffic flows. In other words, the requesting node services the TCP traffic flow first, and then add the request that specifies the required bandwidth of the TCP traffic flow into the head of the MSH-DSCH message to be sent. Upon receiving the MSH-DSCH message, the granting node retrieves a request in the order of request appending sequence in the MSH-DSCH message, and then assigns the schedules for this request. In other words, the granting node will assign the first schedule to TCP traffic flow. As a result, in the “SMGR” and “SMR,” the TCP traffic flow can be serviced best among all the traffic flows.

Second, in the “SMG” and “Original,” which the SMR scheme is not used, we specify that the TCP traffic flow is the first selected traffic flow during every round-robin run. As explained above, the TCP traffic flow can be served more smoothly

than other traffic flows. Thus, in the “SMG” and “Original,” the TCP traffic flow can obtain the highest service quality among all the traffic flows. From above two aspects, we can see that the TCP traffic flow can be serviced best because of its highest priority. As a result, the gaps of TCP performances among the four schemes are very small. Note that the priority values of these traffic flows can be specified flexibly in our design.

Besides, for some traffic flows (e.g., the TCP and CBR with 10 kbyte/sec traffic flows), the achieved application throughputs when the SMG scheme is not used is better than that when the SMG scheme is used. These minor discrepancies can be attributed to the unfairness of resource sharing, which has been explained in Section 6.1.4. Overall speaking, the SMGR scheme has a great improvement in application performances of the end-to-end traffic flows.

6.2.3 Bandwidth Reservation

In this section, we evaluate the support of soft-QoS requirement when using the four schemes, which have been described in Section 6.2.2. We choose the “bandwidth reservation” as our performance metric. It is one of the main properties in the QoS requirement. In the simulation setting, the bandwidth reservation for each TCP traffic is assigned 100 Kbyte/sec, for each UDP traffic is assigned 400 Kbyte/sec, for each CBR with 100 Kbyte/sec traffic is assigned 25 Kbyte/sec and for each CBR with 10 Kbyte/sec traffic is assigned 3 byte/sec.

The used network configuration is shown in Fig. 6.6. We run the four different schemes based on this configuration 3 times, each time using a different random number seed. The simulated time of each run is 200 seconds.

As shown in Table 6.4, the SMGR scheme outperforms the original scheme on the bandwidth reservation achievement ratio of TCP, greedy UDP, CBR with 100 Kbyte/sec and CBR with 10 Kbyte/sec by a factor of 1.40, 2.79, 3.05 and 1.47, respectively. Thus, our proposed scheme can better support the QoS requirement in terms of bandwidth reservation. This is because the bandwidth utilization is more

Table 6.3: MAC-layer and Application-layer performances when using four different schemes

(a) MAC-layer performance

	MAC	
	$M_{Granting}$	M_{Util}
SMGR	44.18	57.00
SMR	29.79	38.36
SMG	26.65	34.58
Original	15.36	19.86

(b) Application-layer performances of TCP and UDP connections

	Application		
	TCP	UDP	
	Throughput (Kbyte/sec)	Throughput (Kbyte/sec)	Packet Delay Time (ms)
SMGR	538.85	913.73	81.28
SMR	549.88	643.86	116.36
SMG	482.10	446.11	173.48
Original	429.23	211.60	373.64

(c) Application-layer performances of CBR with 100 Kbyte/sec and CBR with 10 Kbyte/sec connections

	Application			
	CBR with 100 Kbyte/sec		CBR with 10 Kbyte/sec	
	Throughput (Kbyte/sec)	Packet Delay Time (ms)	Throughput (Kbyte/sec)	Packet Delay Time (ms)
SMGR	44.09	51.07	7.86	113.09
SMR	34.19	119.64	7.97	74.22
SMG	15.65	286.26	3.68	718.15
Original	9.83	484.94	4.04	597.55

Table 6.4: Achievement ratio of requested bandwidth reservation for different traffic flows

(a) TCP and UDP connections

	TCP			UDP		
	Requested B.R. (Kbyte/sec)	Obtained Thro. (Kbyte/sec)	Achievement Ratio (%)	Requested B.R. (Kbyte/sec)	Obtained Thro. (Kbyte/sec)	Achievement Ratio (%)
SMGR	100.00	97.52	97.52	400.00	396.28	99.07
SMR	100.00	95.79	95.79	400.00	373.25	93.31
SMG	100.00	79.38	79.38	400.00	225.84	56.46
Original	100.00	69.75	69.75	400.00	142.09	35.52

(b) CBR with 100 Kbyte/sec and CBR with 10 Kbyte/sec connections

	CBR with 100 Kbyte/sec			CBR with 10 Kbyte/sec		
	Requested B.R. (Kbyte/sec)	Obtained Thro. (Kbyte/sec)	Achievement Ratio (%)	Requested B.R. (Kbyte/sec)	Obtained Thro. (Kbyte/sec)	Achievement Ratio (%)
SMGR	25.00	20.20	80.80	3.00	2.60	86.67
SMR	25.00	23.34	93.36	3.00	2.99	99.67
SMG	25.00	9.34	37.36	3.00	1.69	56.33
Original	25.00	6.63	26.52	3.00	1.77	59.00

efficient.

One sees that when the SMR scheme is used the bandwidth reservation achievement ratio is significantly enhanced. This is because the SMR scheme can simultaneously issue at most four bandwidth requests, each of which is for a specific traffic flow, during a three-way handshake procedure. Thus, not only the network bandwidth utilization can be improved but also the packet delay can be significantly reduced.

Besides, for some traffic flows, the bandwidth reservation achievement ratio when the SMG scheme is not used is better than that when the SMG scheme is used. The reasons for this phenomenon have been explained before.

Chapter 7

Future Work

- Appropriate granting threshold assignment

The SMG scheme provides great improvements on bandwidth utilization at the MAC layer and network capacity at the Application layer. Our simulation results show that the SMG scheme works better than the original scheme under any case. In some conditions, however, ones see that the network performance declines when a very large G_{Thres} value is used. This is because the unfairness of resource sharing problem may occur. Therefore, a more suitable G_{Thres} value should be assigned according to different conditions (i.e., it can be assigned by a dynamic scheme).

- Flexible bandwidth requests for different traffic flows

The SMR scheme can simultaneously issue bandwidth requests for each traffic flow during a three-way handshake procedure. Based on our current design, however, the requested bandwidths for each traffic flow are assigned a fixed value. For different traffic flows, we should serve them by specifying different “shape” requests. For example, we can assign a larger value of frame size and a smaller value of minislot size for TCP. By this way, the TCP can be serviced more smoothly. As a result, by flexibly assigning the bandwidth requests for each traffic flow, the traffic flows are likely to be more smoothly served, and thus the network bandwidth can be more efficiently utilized.

Chapter 8

Conclusion

In an IEEE 802.16 Mesh network, the MAC-layer minislot utilization is very important for achieving good network performances in the distributed coordinated scheduling mode. In this thesis, we show that when the original scheme is used, the Application-layer traffic flows suffer from a very large delay due to the inefficient network bandwidth utilization, and thus the application performances are poor. Besides, the end-to-end traffic flows are served very poorly, even the traffic starvation problem may occur.

To address these problems, we propose a novel scheduling scheme, namely the slicing-based scheduling scheme, to (1) improve MAC-layer scheduling performances and (2) provide better soft-QoS support. Both of the multi-grant mechanism and the multi-request mechanism in the slicing-based scheme are proposed and their performances are studied and compared with the original scheme in this thesis.

The overall simulation results show that the slicing-based scheme significantly outperforms the original scheme. The $M_{Granting}$ and M_{Util} results show that our proposed scheme significantly increases the MAC-layer minislot utilization. In addition, the throughput results show that it generates higher TCP and UDP throughputs than the original scheme. The packet delay results show that the slicing-based scheme results in a shorter end-to-end packet delay than the original scheme. Finally, the Application-layer performances of the end-to-end traffic flows and band-

width reservation achievement ratio results show that our proposed scheme can better support soft-QoS than the original scheme.



Bibliography

- [1] 802.16d 2004. Draft IEEE standard for local and metropolitan area networks - Part 16: Air interface for fixed broadband wireless access systems. May. 2004.
- [2] M. Cao, W. Ma, Q. Zhanq, and X. Wanq. Analysis of IEEE 802.16 Mesh Mode Scheduler Performance. In *IEEE Transactions on Wireless Communications*, pages 1455–1464, April. 2007.
- [3] H. Chu. Simulating WiMAX Mesh Networks over the NCTUns Network Simulator. June. 2003.
- [4] L. Fu, Z. Cao, and P. Fan. Spatial Reuse in IEEE 802.16 Based Wireless Mesh Networks. In *IEEE International Symposium on Communications and Information Technology 2005 (ISCIT 2005)*, pages 1358–1361, Oct. 2005.
- [5] J. Tao, F. Liu, Z. Zanq, and Z. Lin. Throughput Enhancement in WiMAX Mesh Networks Using Concurrent Transmission. In *Proceedings of International Conference on Wireless Communications, Networking and Mobile Computing, 2005, WCNM 2005*, pages 871–874, Sept. 2005.
- [6] S. Wang, C. Chou, C. Huang, C. Hwang, Z. Yang, C. Chiou, and C. Lin. The Design and Implementation of the NCTUns 1.0 Network Simulator. *Computer Networks*, 42(2):175–197, June. 2003.
- [7] S. Wang, C. Lin, K. Fang, and T. Hsu. Facilitating the Network Entry and Link Establishment Processes of IEEE 802.16 Mesh Networks. In *Wireless Communication and Networking Conference 2007 (WCNC 2007)*, Mar. 2007.