國立交通大學

資訊科學與工程研究所

博士論文

增進 IEEE 802.16 網狀分散協同排程網路的效能

Performance Enhancements of the IEEE 802.16 Mesh Coordinated Distributed Scheduling Mode Network

研 究 生：林志哲

指導教授：王協源　教授

中華民國九十九年七月

增進 IEEE 802.16 網狀分散協同排程網路的效能
Performance Enhancements of the IEEE 802.16 Mesh Coordinated Distributed Scheduling Mode Network

研究生：林志哲　　Student：Chih-Che Lin

指導教授：王協源 博士　　Advisor：Dr. Shie-Yuan Wang

國立交通大學
資訊科學與工程研究所
博士論文

A Dissertation
Submitted to Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy
in

Computer Science

July 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年七月

增進 IEEE 802.16 網狀分散協同排程網路的效能

學生：林志哲　　　　　　　　　　　　　　　　　　　　　　指導教授：王協源教授

國立交通大學

資訊科學與工程研究所

摘　　要

無線網狀網路為新一代可節省成本的骨幹傳輸網路 (Backbone Relay Network) 解決方案。而 IEEE 802.16 網狀分散協同排程網路 (Mesh Coordinated Distributed Scheduling Network)為下一世代無線網狀網路的候選方案之一。此一新世代的網路具有「無封包碰撞」的特性，與傳統的以 802.11 為基礎的無線網狀網路有非常大的差異。

在此篇論文裡，我們提出了數個提升此一新網路效能的方案，包括了提升其控制訊息排程效率的方案與增加其服務品質 (Quality of Service) 的方案。除此之外，針對此網路的初始化流程。我們發現了一個網路節點在任意拓墣 (General Topology) 裡，可能會初始化失敗的問題，並且提出解決的方案。在此篇論文裡，我們以理論分析與模擬的方式來評估所提出的方案的效能。我們的理論分析與模擬結果皆顯示我們所提出的各項方案能有效地提升 IEEE 802.16 網狀分散協同排程網路的效能。

關鍵字：無線網狀網路、802.16、分散式排程

Performance Enhancements of the IEEE 802.16 Mesh Coordinated Distributed Scheduling Mode Network

Student: Chih-Che Lin　　　　Advisor: Dr. Shie-Yuan Wang

Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University

ABSTRACT

The Wireless Mesh Network (WMN) is a cost-effective solution for backbone networks in both metropolitan and rural areas and can be used as temporary broadband access for emergent and tactic purposes. Without the need of wires, WMNs are easy to deploy and reconstruct to satisfy the dynamic needs. The IEEE 802.16 mesh CDS-mode network is a candidate of next-generation WMNs, which provides the "collision-free" property unique to traditional IEEE 802.11 based WMNs.

In this dissertation, we propose several schemes for this new network to enhance its scheduling efficiency on the control plane and its QoS support on the data plane. In addition, we also point out the issues of network initialization of this network on random topologies and propose a scheme to solve these issues. The performances of our proposed schemes are evaluated using both analyses and simulations. Our analytical and numerical results show that our proposed schemes can significantly enhance the performance of the IEEE 802.16 mesh CDS-mode network and benefits upper-layer applications.

Keywords: 802.16, wireless mesh network, distributed scheduling

致謝

回顧我的求學生涯，已在交通大學裡待了十二個年頭。這一路艱辛的研究旅程，我要感謝指導教授王協源老師的悉心栽培。在他的教導之下，我於研究、思考、專案管理以及文章撰寫都逐漸地成熟。我也要感謝我的校內校外論文指導委員（林一平院長、曾煜棋副院長、曾建超所長、清華大學許建平教授、成功大學郭耀煌教授，以及台灣大學林風教授等）給予我的論文具有建設性的建議，使我能在畢業之際發掘自己的缺點，加以改進，成為一個更加成熟的研究者。我亦要感謝所有在實驗室裡同甘共苦的周智良學長與歷屆學長學弟妹，沒有你們的共同參與及集思廣益，我的許多研究將不會那麼順利完成。

在這漫長的研究歷程，我要感謝我父親與母親的支持與體諒。感謝他們無私地支持著我完成這一個漫長的學業歷程，感謝他們即便處於最艱困的時刻都仍努力成為我生活上的後盾。沒有他們在生活上的全力支持，容忍我將人生的第一個黃金階段花在十年的學業上，就沒有今天取得博士學位的我。

此外，我要感謝我的外公與各位阿姨，感謝你們這麼多年來默默地支持我與母親，使得我能在這一生活艱困的時刻，仍能心無旁騖於研究與學位。沒有你們的支持，我的家庭無法支撐到我畢業的這一刻。

最後，我要感謝我的女友莉晶，在我鬥志最黯淡低落的時刻，總是陪在我的身旁，照顧我的生活，給予我最大的鼓勵，讓我能度過一次次的低潮。我亦要感謝莉晶的母親，感謝您透過莉晶給予我們倆生活的支持，使我能在最後的關鍵階段能放心地在埋首在研究上，以完成學業。

林志哲 撰於 民國九十九年七月二十七日

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Wireless Mesh Network (WMN) is a cost-effective solution for backbone networks in both metropolitan and rural areas and can be used as temporary broadband access for emergent and tactic purposes. Without the need of wires, WMNs are easy to deploy and reconstruct to satisfy the dynamic needs. The IEEE 802.16 mesh network is a candidate of next-generation WMNs in this decade, which uses a TDMA-based Medium Access Control (MAC) layer and OFDM-based physical layer and mainly operates at a single frequency. In this network, packets can be transferred in a peer-to-peer manner, and network accesses are managed in a TDMA-like fashion.

As shown in Fig. 1.1, network bandwidth is first divided into frames, each of which is subdivided into one control and one data subframes. A control subframe is further divided into Transmission Opportunities (TxOpp) while a data subframe is further divided into minislots. Control messages and data packets are transferred over TxOpps and minislots, respectively.

To avoid conflicts in using TxOpps, the IEEE 802.16 mesh network defines two scheduling modes: 1) the Centralized Scheduling (CS) mode and 2) Distributed Scheduling (DS) modes. In the CS mode, a network is partitioned into tree-based clusters. Each cluster has a Base Station (BS) node responsible for allocating network resources to the Subscriber Station (SS) nodes that it services. Although the CS mode provides collision-free transmissions for control messages and data, it has several disadvantages described below.

First, the number of routes that can be utilized is unnecessarily reduced. The reason is that the CS mode uses a tree-based topology, which cannot exploit all possible routes in a network, as compared with a mesh-based topology. For the same reason, the only route

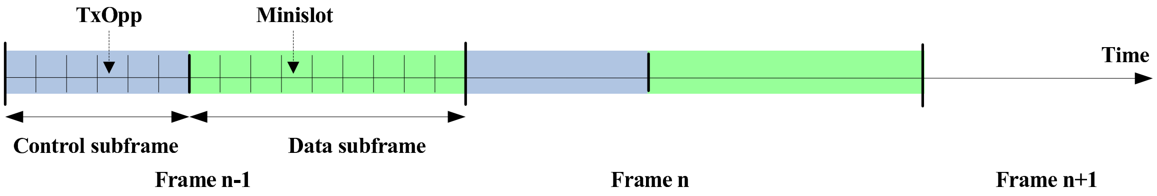


Figure 1.1: The frame structure of the IEEE 802.16 mesh mode

between two nodes on the tree may not be the shortest one between them if instead a mesh-based topology were used. In addition, the root node of the tree is likely to become the performance bottleneck because many packets need to pass through it to reach their destination nodes.

Second, it is difficult to efficiently exploit the spatial reuse property of wireless communication in the CS mode. The message format defined in this mode only allows a BS node to notify an SS node of the bandwidth allocated for it. There is no field in the message to allow a BS node to specify the start and end minislot offsets for an allocation. Thus, to avoid interference, each SS node has to take a conservative approach to derive its own data schedule. Allocating minislots in this way is collision-free but results in only one active SS node per cluster at any given time. More detailed explanations about this problem are provided in Appendix.

In contrast, the DS mode provides two advantages. First, the DS mode uses a mesh topology. This allows all possible routing paths to be utilized to avoid performance bottlenecks. Besides, spatial reuse of wireless communication can be exploited to increase network capacity. Second, the DS mode establishes data schedules on an on-demand basis and thus network bandwidth can be more efficiently utilized.

In the IEEE 802.16 mesh network standard, the DS mode is further divided into two operational modes: the coordinated and uncoordinated modes. In the Coordinated Distributed Scheduling (CDS) mode, the control messages required for scheduling minislot allocations are transmitted over TxOpps without collisions. In contrast, in the Uncoordinated Distributed Scheduling (UDS) mode, such control messages can only be transmitted on the TxOpps left from the CDS mode or on unallocated minislots. As a result, the CDS mode provides better QoS supports than the UDS mode. In this dissertation, we focus on the 802.16 mesh CDS mode. Fig. 1.2 shows an example IEEE 802.16 mesh CDS-mode network with 25 nodes, where the BS node is at the center of the network.

In the 802.16 mesh CDS mode, each node uses the same pseudo-random election al-

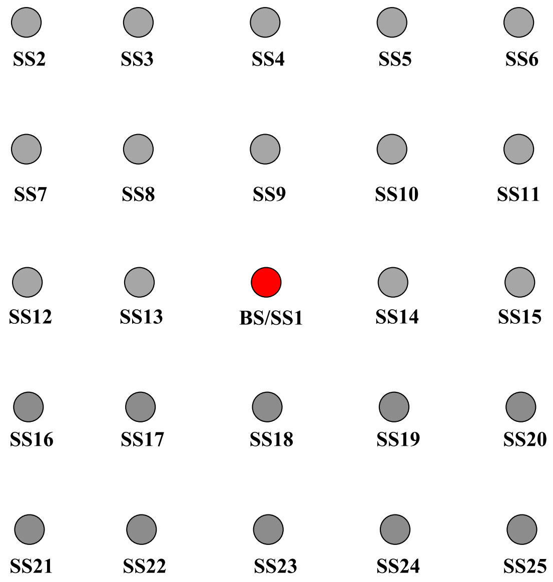


Figure 1.2: Example of the IEEE 802.16 mesh CDS-mode network

gorithm [1] to resolve contention for TxOpps. Control messages, such as mesh network configuration (MSH-NCFG) and mesh distributed coordinated scheduling (MSH-DSCH) messages, are transmitted on TxOpps determined by this algorithm. (The MSH-NCFG message is used to carry information for network initialization and the MSH-DSCH message is used to carry information for minislot scheduling.) The contention resolution of MSH-DSCH message transmissions is explained below to illustrate how this pseudo-random algorithm works.

During operation, each node listens to MSH-DSCH messages advertised by its neighboring nodes. Based on the scheduling information carried in MSH-DSCH messages, each node knows the transmission interval (shown in Fig. 3.4 and will be explained in detail later) of each of its neighboring nodes. More specifically, it knows for which TxOpps its neighboring nodes may contend. Each node then uses the same election algorithm to determine the winning node for a given TxOpp. This algorithm takes a specified TxOpp number and the IDs of all nodes contending for this TxOpp as input. It outputs the ID of the winning node whose computed value is the largest among all the competing nodes. Conceptually, every node uses the same algorithm and the same input (regarding its own two-hop neighborhood). It thus knows which node will win a given TxOpp within its two-hop neighborhood. As a result, no message collisions will occur on any TxOpp. If a node cannot win a given TxOpp, it repeats the above process with the next TxOpp (i.e., the previous TxOpp number plus one) as input until it eventually wins one TxOpp.

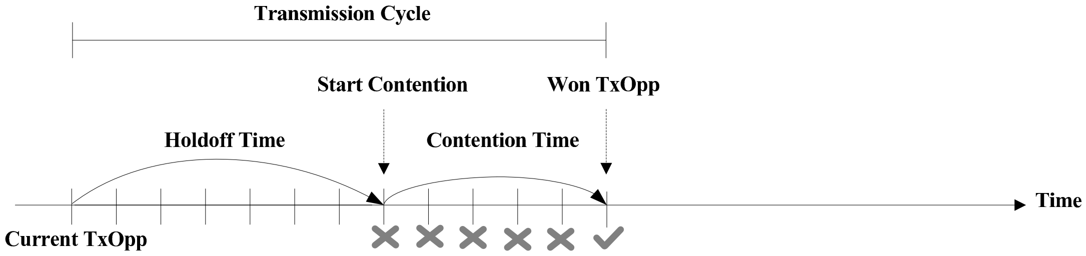


Figure 1.3: A node's transmission cycle comprises the holdoff time and the contention time.

After a node wins a TxOpp, the IEEE 802.16 mesh CDS mode requires it to refrain from contending for another TxOpp in a certain number of consecutive TxOpps (called the holdoff time). As shown in Fig. 1.3, a node's transmission cycle comprises the holdoff time and the contention time. The contention time is defined as the number of consecutive TxOpps in which a node should contend for access (i.e., participate in the algorithm computation) until it wins one. Using this design, every winning node has to suspend its contention for its next MSH-DSCH TxOpp during its holdoff time. Thus, a winning node cannot obtain more than one TxOpp in one transmission cycle, preventing a node from monopolizing a wireless channel. The holdoff time design allows for fair accesses to TxOpps among competing nodes at a cost of increasing delays in transmitting control messages.

The effect of the holdoff time value can be discussed from two aspects. On one hand, if the holdoff time is set to a too large value, network nodes will suffer from long delays in transmitting control messages and thus cannot fully utilize the link bandwidth. In the CDS mode, a network node can transmit data only after it has scheduled a minislot allocation with the intended receiving node. This requires the node and its intended receiving node to perform a Three-way Handshake Procedure (denoted as THP) [1]. THP requires transmitting three MSH-DSCH messages to exchange necessary minislot scheduling information, which means that it requires three MSH-DSCH TxOpps to complete the handshake process. Since using a large holdoff time value will increase the transmission cycle for transmitting a control message, the time required for handshaking a minislot allocation (and thus transmitting data) will also be increased.

On the other hand, if the holdoff time is set to a too small value, the number of nodes competing for a TxOpp will be large. This will lead to high contention for TxOpps. In such a congested condition, nodes may experience unpredictable packet delays and share

TxOpps unfairly. In summary, the holdoff time should be set to an appropriate value that is large enough to avoid congestion but small enough to avoid large transmission delays.

The 802.16 mesh CDS mode uses a fixed holdoff time design in which the holdoff time of each node is a fixed value greater than or equal to 16. This fixed-value design cannot perform optimally for all networks because it does not consider the number of competing nodes and traffic load may vary in the network. To enhance the performances of the 802.16 mesh CDS mode, in this dissertation we propose dynamic holdoff time setting designs for IEEE 802.16 mesh CDS-mode networks using omnidirectional antennas and those using single-switched-beam antennas.

1.1 Problem Description

The IEEE 802.16 mesh CDS mode uses a fixed holdoff time design to schedule the control message transmissions of nodes. This design guarantees the control message transmissions in the network to be collision-free; however, due to the fixed holdoff time, it is inflexible to adjust nodes' control message transmissions when network traffic dynamically changes. Thus, the IEEE 802.16 mesh CDS mode cannot achieve the most efficient control and data scheduling at all time.

To enhance the performances of this new network, in this dissertation we propose a dynamic holdoff time design that can efficiently schedule the transmissions of control messages and data in the mesh CDS mode network based on dynamic traffic needs. The proposed dynamic holdoff time design can be used in both networks using omnidirectional-antennas and those using directional antennas. The analytical and numerical results show that our proposed dynamic holdoff time design can significantly enhance the performances of the IEEE 802.16 mesh CDS-mode network.

1.2 Dissertation Organization

The remainder of this dissertation is organized as follows. In Chapter 2, we introduce the operation of the IEEE 802.16(d) mesh CDS mode in detail. In Chapter 3, we explain the proposed dynamic holdoff time designs for IEEE 802.16 mesh CDS-mode networks using omnidirectional antennas and single-beam antennas. In Chapter 4, the performances of the proposed dynamic holdoff time designs are evaluated using both analytical models and simulations. In Chapter 5, we discuss the enhancements to the IEEE 802.16 mesh CDS mode in the network initialization process and the minislot handshake process. These enhancements are orthogonal to our main work; however, they significantly affect the performances of this network. We then present related work in Chapter 6. We finally conclude this dissertation and present our future work in Chapter 7.

Chapter 2

Introduction to the IEEE 802.16(d) Mesh CDS Mode

In the IEEE 802.16(d) Mesh CDS-mode network, link bandwidth is divided into frames on the time axis and managed in a time-division-multiple-access (TDMA) manner. Each frame comprises one control and one data subframes. A control subframe is further divided into TxOpps, whereas a data subframe is further divided into minislots. Control messages and data packets are transmitted over TxOpps and minislots, respectively.

The 802.16 mesh CDS mode defines three types of control messages: 1) Mesh Network Entry (MSH-NENT); 2) Mesh Network Configuration (MSH-NCFG); and 3) Mesh Distributed Coordinated Scheduling (MSH-DSCH). The MSH-NENT and MSH-NCFG messages are used for nodes to exchange control information for network initialization whereas the MSH-DSCH message is used for nodes to schedule minislot allocations. A minislot allocation is a set of consecutive minislots across several consecutive data subframes for data packet transfers. The mesh CDS-mode standard categorizes TxOpps into three types, each for transmitting a specific type of control messages. The TxOpps used for transmitting MSH-NENT messages are called MSH-NENT TxOpps; those used for transmitting MSH-NCFG messages are called MSH-NCFG TxOpps; and those used for transmitting MSH-DSCH messages are called MSH-DSCH TxOpps. The detailed usages of these TxOpps are defined in [1].

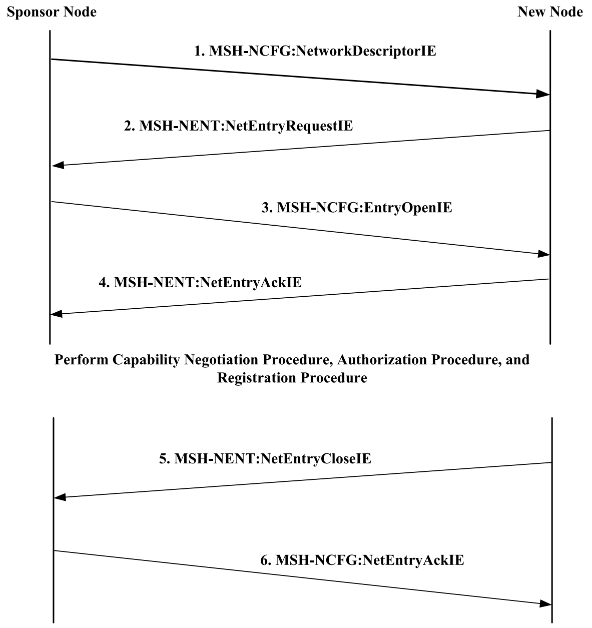

Sponsor Node
New Node
1. MSH-NCFG:NetworkDescriptorIE
2. MSH-NENT:NetEntryRequestIE
3. MSH-NCFG:EntryOpenIE
4. MSH-NENT:NetEntryAckIE
Perform Capability Negotiation Procedure, Authorization Procedure, and Registration Procedure
5. MSH-NENT:NetEntryCloseIE
6. MSH-NCFG:NetEntryAckIE

Figure 2.1: The procedure of the network entry process

2.1 Network Initialization

To join an IEEE 802.16(d) mesh network, an SS must perform the "network entry" process to attach itself to the network. An SS cannot transmit and receive data packets until it finishes its network entry process. As shown in Fig. 2.1, in the network entry process a new SS first monitors the MSH-NCFG messages broadcast from its neighboring operational nodes. (A operational node is an SS that has completed its network entry process.) Based on the received MSH-NCFG messages, this new node builds a neighboring node list that records the ID and operational information of its neighboring operational nodes.

With this list, the new node can start attaching itself to the network. It first selects one of its neighboring operational nodes as its sponsoring node. Then it transmits a network entry request (NetEntryRequest) information element (IE), which is carried in an MSH-NENT message, to the chosen sponsoring node. Upon receiving such a message, the sponsoring node determines whether it can (and is willing to) service this new node. If not, it ignores this message. Otherwise, it reserves a temporary minislot allocation for the new node and then responds the new node with an MSH-NCFG message that carries an EntryOpenIE.

When receiving the MSH-NCFG:EntryOpenIE message, the new node first acknowledges its sponsoring node with a NetEntryAckIE (which is carried in an MSH-NENT message) and then performs the capability negotiation, authorization, and registration procedures with a BS node in the network using the minislot allocation temporarily reserved by the sponsoring node for it. After finishing the registration procedure, the new node has attached itself to the network.

The next step is to transmit a NetEntryCloseIE (which is carried in an MSH-NENT message) to its sponsoring node. The NetEntryCloseIE is used to notify the sponsoring node that the new node has joined the network and thus the temporary minislot allocation is no longer required. Upon receiving this message, the sponsoring node cancels the temporary minislot allocation reserved for the new node and replies the new node with a NetEntryAckIE (which is carried in an MSH-NCFG message), to terminate this sponsorship. Upon receiving the MSH-NCFG:NetEntryAckIE message, the new node finishes its network entry process and becomes a operational node in the network.

2.2 Control Message Scheduling

In the 802.16 mesh CDS mode, each node's control message transmission is scheduled by the Mesh Election Algorithm (MEA) [1], which is hash-based algorithm with an exponential holdoff mechanism. In this mode, each node needs to maintain (i.e., learn) the information about its two-hop neighborhood (which is an input of MEA). The two-hop neighborhood of a node is defined as the set comprising the node itself, its one-hop neighboring nodes, and its two-hop neighboring nodes, i.e.,

$$nbr(i) = \{i\} \cup \ nbr_1(i) \cup \ nbr_2(i), \tag{2.1}$$

where $nbr_1(i)$ and $nbr_2(i)$ are defined as follows, respectively.

$$\begin{aligned} nbr_1(i) = \ & \{k \mid \text{node } k \in \text{node } i\text{'s one-hop neighboring nodes,} \\ & \qquad \text{when omnidirectional antennas are used.}\} \end{aligned} \tag{2.2}$$

$$\begin{aligned} nbr_2(i) = \ & \{k \mid \text{node } k \in \text{node } i\text{'s two-hop neighboring nodes,} \\ & \qquad \text{when omni-directional antennas are used.}\} \end{aligned} \tag{2.3}$$

The purpose of maintaining the two-hop neighborhood is to avoid the hidden terminal problem when transmitting control messages. By using the same MEA in every node, this algorithm guarantees that in the network only one node in any two-hop neighborhood will win the access to a specific TxOpp. Thus, when nodes transmit their control messages, no message collisions will occur. Since the transmission of control messages are collision-free and these messages are used for scheduling a collision-free minislot allocation, the transmission of data can also be guaranteed collision-free.

The scheduling process of the MSH-DSCH message in the 802.16 mesh CDS-mode network is explained here[1]. Each node should perform MEA to determine the TxOpp on which to broadcast its next MSH-DSCH message. MEA takes a given TxOpp number and an eligible node list (which is a list of nodes that are eligible to contend for the given TxOpp) as input. It then iteratively computes a hash value for each node in the given eligible node list on the give TxOpp number. The hash function is given in [1]. Finally, it outputs the ID of the winning node whose computed hash value i s the largest among all of the competing nodes. Because nodes within two hops use the same MEA and consistent eligible node lists, every node knows which node will win a given TxOpp within its two-hop neighborhood.

To achieve the consistency of neighboring nodes' eligible node lists for each TxOpp, each node should periodically broadcast the next MSH-DSCH TxOpps used by itself and its one-hop neighboring nodes. In addition, for node i, if the information about the next MSH-DSCH message transmission of its neighboring node j is unknown, node i will conservatively consider that node j will contend for every TxOpp and put node j into the eligible node list for the following TxOpps until it receives the information about the next MSH-DSCH TxOpp used by node j. Using this design, no message collision will occur on any TxOpp. If a node cannot win a given TxOpp, it repeats the above process with the next TxOpp (i.e., the previous TxOpp number plus one) as input until it eventually wins one TxOpp.

The eligibility of a node for contending for a TxOpp is determined by the holdoff time mechanism [1]. The holdoff time mechanism first defines the control message transmission cycle of a node as the time interval between the node's two consecutive control message transmissions. Recall that, as shown in Fig. 2.2, the transmission cycle of a node comprises 1) the holdoff time and 2) the contention time. The former is defined as the number of consecutive TxOpps during which a node must suspend its contention for TxOpps after

[1] The scheduling process of the MSH-NCFG message is the same as that of the MSH-DSCH message. Thus, we only explain the latter in this chapter for brevity.

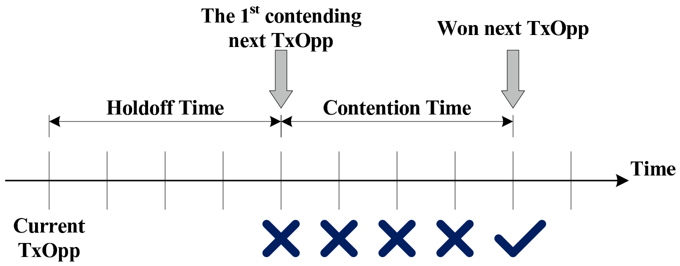


Figure 2.2: The transmission cycle of a node

wining a TxOpp, while the latter is defined as the number of consecutive TxOpps for which a node may contend to win a TxOpp. Thus, by obtaining the holdoff times of the nodes in its two-hop neighborhood, a node can know for which TxOpps these neighboring nodes will and will not contend. Based on such information, it can construct an eligible node list of its two-hop neighborhood for every TxOpp.

In the 802.16(d) mesh CDS mode, the holdoff time of each node is fixed and defined as follows:

$$\mathrm{HT}(i) = 2^{exp+base}, \tag{2.4}$$

where HT(i) denotes the holdoff time of node i. The base value is fixed to 4 and the range of the *exp* value is between 0 and 7. In the 802.16 mesh mode standard, the *exp* value of each node is the same. Thus, the holdoff times of all nodes in the network are the same. In contrast, the contention times of nodes may vary depending on which and how many nodes are eligible to contend for TxOpps.

Before broadcasting an MSH-DSCH message, a node first executes MEA to calculate (win) a TxOpp for its next MSH-DSCH message transmission. It then puts the following information into the MSH-DSCH message that is going to be sent out: 1) its calculated next MSH-DSCH TxOpp number; 2) its holdoff time; 3) the learned next MSH-DSCH TxOpp numbers of its one-hop neighboring nodes; and 4) the holdoff times of these one-hop neighboring nodes. After this, it broadcasts this MSH-DSCH message to all of its neighboring nodes.

By this design, a node can learn in advance the next MSH-DSCH TxOpp numbers and holdoff times of all nodes in its two-hop neighborhood. Because every node knows such information about every other node in its two-hop neighborhood, for any given TxOpp, the output of every node's MEA in any two-hop neighborhood is consistent. The consistency of the MEA output means that: For nodes i and j, suppose that $\{a, b\} = nbr(i) \cap nbr(j)$,

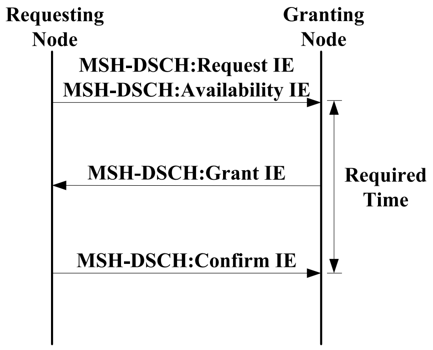


Figure 2.3: The THP defined in the IEEE 802.16(d) mesh CDS mode

it is impossible that a and b simultaneously win the same TxOpp. That is, in a two-hop neighborhood, there is only one winning node for a given TxOpp and every node knows who wins it. As a result, when nodes transmit their MSH-DSCH messages, no collision will occur. It is this advance announcement design that enables MEA to generate collision-free control message scheduling.

Another advantage of the holdoff time design used in the 802.16 mesh CDS mode is that, using this design, a node must refrain from contending for TxOpps after winning a TxOpp (i.e., after transmitting its current MSH-DSCH message) until its holdoff time has elapsed. This ensures that nodes other than the winning node will have a chance to win subsequent TxOpps and all nodes can fairly share TxOpps in the long run.

2.3 Data Minislot Scheduling

The IEEE 802.16(d) mesh CDS mode schedules data transmissions of nodes in a distributed and on-demand manner. A three-way handshake procedure (THP) [1] is used for a pair of nodes to negotiate a minislot allocation agreed by both nodes. A minislot allocation is a set of consecutive minislots on which the transmitting node and the receiving node are sure that they are ready to transmit and receive data packets. As shown in Fig. 2.3, THP uses a “request-grant-confirm” control sequence for two peer nodes to negotiate a minislot allocation. In a THP, the requesting node first transmits a request IE and an availability IE to the granting node using an MSH-DSCH message. The request IE specifies the number of minislots that the requesting node needs to transmit data packets and the availability IE specifies a set of consecutive minislots on which the requesting node can transmit data packets.

On receiving a request IE, the granting node first determines whether it can receive data packets from the requesting node within the minislot set specified by the received availability IE. If not, the granting node can simply ignore the received request IE. Otherwise, it allocates a minislot allocation within the specified minislot set and then transmits a grant IE to the requesting node as an acknowledgment using its MSH-DSCH message. The grant IE specifies a subset of the minislots specified by the received availability IE on which the granting node is willing to receive data packets from the requesting node. Upon receiving the grant IE, the requesting node then broadcasts a confirm IE using its MSH-DSCH message to complete this THP. The confirm IE is a copy of the received grant IE and used to notify the requesting node's one-hop neighboring nodes of the duration on which this minislot allocation will take place.

2.4 Acronym Table

Many acronyms are used in this dissertation to make it concise. To help readers easily refer to the meanings of these acronyms, frequently-used acronyms are collected and defined in Tab. 2.1.

Table 2.1: Frequently-used Acronyms

Acronym	**Meaning**
ATUF	Average Throughput of a UDP Flow in a network
ATTF	Average Throughput of a TCP Flow in a network
ATHPT	Average Three-way Handshake Procedure Time
ATOUN	Average Transmission Opportunity Utilization of Nodes
BS	Base Station
CDS	Coordinated Distributed Scheduling
CS	Centralized Scheduling
CV-ATUF	The Coefficients of Variation of all nodes' ATUF values
CV-ATTF	The Coefficients of Variation of all nodes' ATTF values
MEA	Mesh Election Algorithm
MEAI	Mesh Election Algorithm Instance
TMEA-S	Transmission-domain-aware Mesh Election Algorithm with a Static holdoff time design
TMEA-D	Transmission-domain-aware Mesh Election Algorithm with a Dynamic holdoff time design
MSH-DSCH	Mesh Distributed Scheduling
MSH-NCFG	Mesh Network Configuration
MSH-NENT	Mesh Network Entry
MTD	Multi-transmission-domain
NetUI	Network Unfairness Index
NodeUI	Node Unfairness Index
STD	Single-transmission-domain
TD	Transmission Domain
TDI	Transmission Domain Index
THP	Three-way Handshake Procedure
TMEA	Transmission-domain-aware Mesh Election Algorithm
TMSA	Transmission-domain-aware Minislot Scheduling Algorithm
TxOpp	Transmission Opportunity
UDS	Uncoordinated Distributed Scheduling

Chapter 3

Proposed Dynamic Holdoff Time Designs

3.1 Proposed Scheme for Networks using Omni-directional Antennas

3.1.1 Motivation

In Section 1, we have explained that the IEEE 802.16 mesh CDS mode uses a fixed holdoff time design to schedule the control message transmissions of nodes. Due to this fixed holdoff time, it is inflexible to adjust nodes' control message transmissions when network traffic dynamically changes. Thus, the IEEE 802.16 mesh CDS mode cannot achieve the most efficient control and data scheduling at all time.

One intuitive alternative is to totally shutdown the control message transmissions of those nodes that do not have data to send. However, as explained in Section 2.2, if a node does not transmit its control messages (i.e., MSH-NCFG and MSH-DSCH messages), based on [1] its neighboring nodes will conservatively consider that this node will contend for every TxOpp. In this condition, the number of contending nodes per TxOpp will be increased (due to the unknown status of the node that disables its control message transmissions).

To solve this problem, the formats of the MSH-NCFG and MSH-DSCH messages should be expanded to allow a node to use a very large holdoff time value (or the infinite holdoff time value). Using such a design, when a node disables its control message

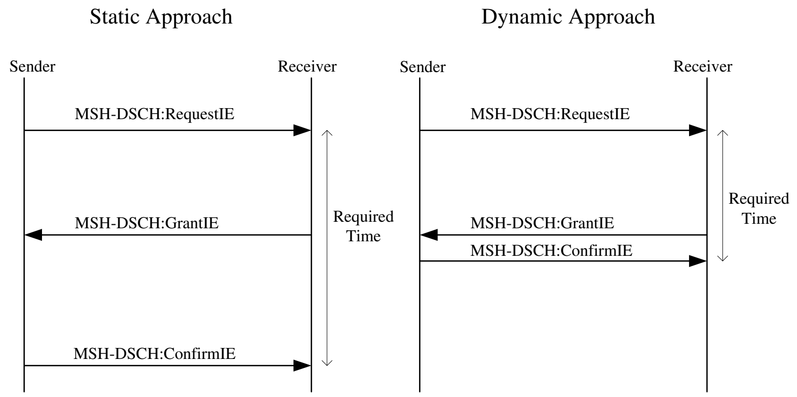


Figure 3.1: The advantage of the proposed dynamic holdoff time scheme

transmissions for a very long period, its neighboring nodes can safely exclude it from the contending node lists of subsequent TxOpps, which can effectively reduce the number of contending nodes per TxOpp. Because this solution requires the expansion of control message formats, we do not adopt this solution to avoid changing the message formats of the IEEE 802.16 mesh CDS-mode specification.

3.1.2 Design

The main idea of the proposed dynamic holdoff time scheme is to decrease the transmission cycles of nodes, when they have data to send. This is accomplished by decreasing the time required by nodes to complete their THPs. By doing this, per-hop (and end-to-end) data transmission delays can be decreased and per-hop (and end-to-end) data transmission throughputs can be increased.

Using the proposed dynamic holdoff time scheme, a requesting node can reduce the time interval between transmitting its request IE and transmitting its confirm IE. That is, using this scheme, on receiving a grant IE from the granting node, a requesting node will transmit a confirm IE as soon as possible. As shown in Fig. 3.1, the proposed dynamic holdoff time scheme can on average save about a half of the time required to establish a minislot allocation, compared with fixed holdoff time schemes.

The detailed algorithm of our proposed dynamic holdoff time scheme is shown in Fig. 3.2 and explained here. In this scheme, the holdoff time base value is set and fixed to 0 rather than the default 4. Initially, a node using the proposed scheme determines its holdoff time using the following formula:

$$\mathrm{HT}(i) = 2^{floor(log_2(|nbr(i)|))} \tag{3.1}$$

where $nbr(i)$ is the number of nodes in node i's two-hop neighborhood and defined in Eq. (2.1). If this node does not have data to send,it regularly transmits its MSH-DSCH messages using Eq. (3.1) to keep MEA operating correctly. Note that even though there is no data to send, a node still needs to send out its MSH-DSCH messages regularly to maintain the operation of its MEA. The transmitted MSH-DSCH messages are used to notify this node's neighboring nodes of its next MSH-DSCH message transmission time. These MSH-DSCH messages, however, need not carry minislot scheduling information, such as request, grant, and confirm IEs.

In contrast, when a node has data to send, it first needs to launch a THP (i.e., transmit a request IE out). In this condition, before the requesting node transmits out the request IE, it calculates the earliest TxOpp where it can transmit the confirm IE to the granting node in advance. Note that transmitting the confirm IE must be performed later than receiving a grant IE from the granting node. To ensure this sequence, the requesting node's target TxOpp (i.e., the next TxOpp used to transmit the confirm IE) is initially set to the next TxOpp of the granting node plus one. Note that the requesting node knows this information because this information is regularly exchanged among nodes via the MSH-DSCH messages.

Then, it uses the difference between its current and the target TxOpps to calculate the maximum target holdoff exponent value as follows:

$$\text{max target holdoff exp} = \lceil log_2(\text{difference}) \rceil \tag{3.2}$$

This calculated exponent value is used as the initial exponent value to find a TxOpp that is later than the target TxOpp. The found TxOpp is the output of MEA and will be larger than the target TxOpp due to the existence of the contention time.

Later on, our proposed dynamic holdoff time scheme then goes through an iteration to find the smallest exponent value that makes the transmission of the confirm IE as close as possible to the reception of the granting IE. During each step of the iteration, the target holdoff time exponent is decremented by one to explore whether this smaller value can still meet the requirement. On the last step of the iteration, where the requirement can no longer be met, the exponent value used in the previous step (which is stored in the

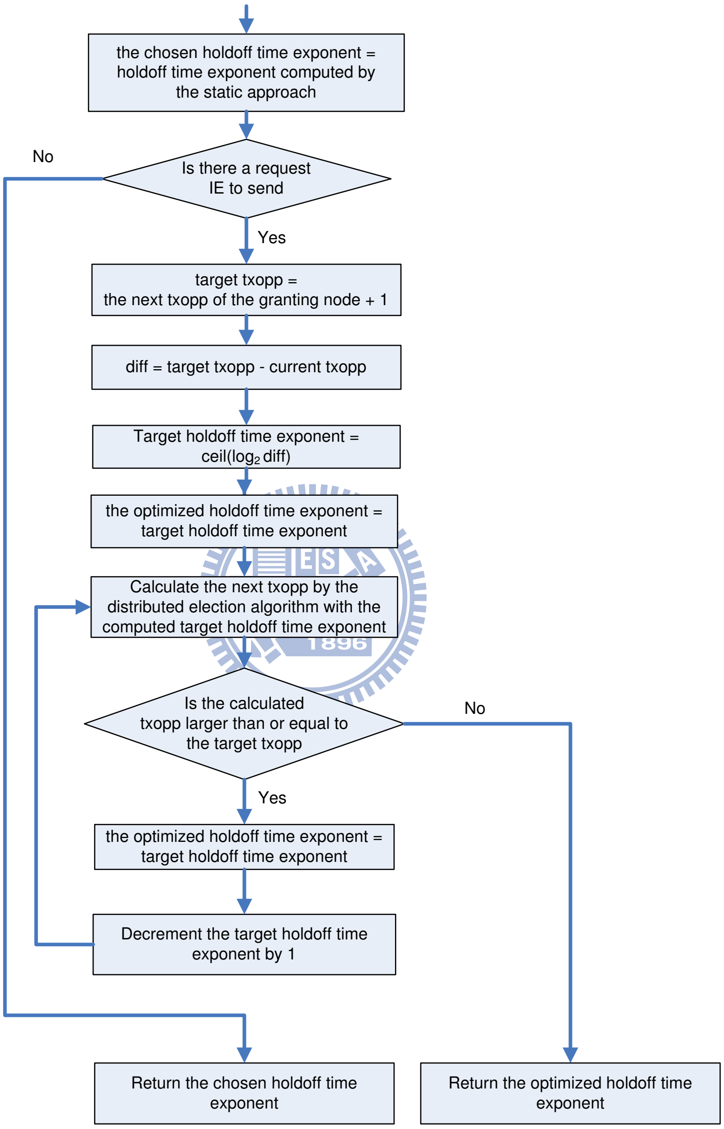

the chosen holdoff time exponent = holdoff time exponent computed by the static approach
Is there a request IE to send
No
Yes
target txopp = the next txopp of the granting node + 1
diff = target txopp - current txopp
Target holdoff time exponent = ceil(log2 diff)
the optimized holdoff time exponent = target holdoff time exponent
Calculate the next txopp by the distributed election algorithm with the computed target holdoff time exponent
Is the calculated txopp larger than or equal to the target txopp
No
Yes
the optimized holdoff time exponent = target holdoff time exponent
Decrement the target holdoff time exponent by 1
Return the chosen holdoff time exponent
Return the optimized holdoff time exponent

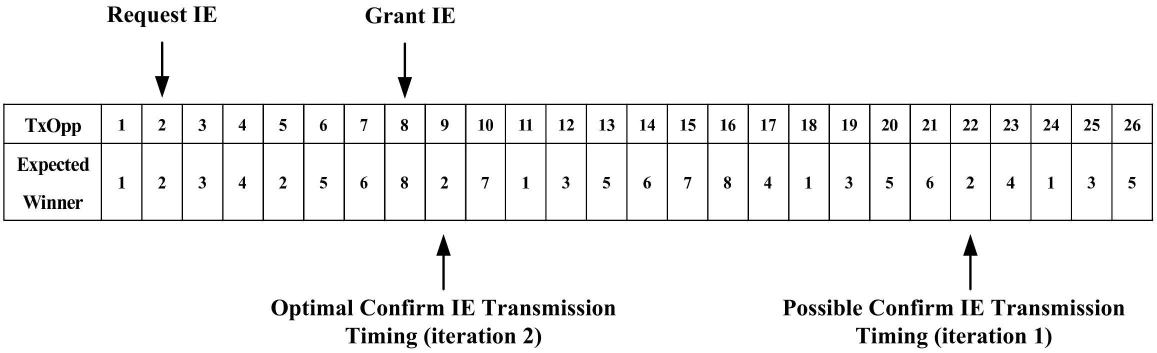
Figure 3.2: The algorithm of the dynamic approach of the proposed scheme

optimized holdoff time exponent variable) is the exponent value that is both feasible and the smallest. This value is then returned and used to derive the TxOpp for transmitting the confirm IE.

Fig. 3.3 shows an example illustrating the operation of the proposed dynamic holdoff time scheme. Suppose that the two-hop neighborhood of node 2 comprises eight nodes, including node 2 itself. Fig. 3.3(a) shows the winning nodes of the TxOpps numbered from 1 to 26. Here, the winning node of a TxOpp is defined as the node that wins this TxOpp when all of the eight nodes contend for this TxOpp. (This condition will occur if the holdoff time base value is set to zero. The effect of the holdoff time base value will be explained in detail in Section 3.1.3.) Assume that node 2 intends to schedule a minislot allocation with node 8. Before transmitting the request IE to node 8 on TxOpp 2, node 2 should perform the algorithm depicted in Fig. 3.2. The detailed steps executed by this algorithm are explained below.

As shown in Fig. 3.3(b), the proposed algorithm first sets the target TxOpp to 9, which is right after the TxOpp that node 8 is likely to transmit its grant IE (assuming 8 in this example). The algorithm calculates the TxOpp that node 2 can win. During the first iteration, the algorithm first finds that node 2 can win TxOpp 22. Since the calculated transmission opportunity (22) is still larger than the target TxOpp, the algorithm stores the current target holdoff time exponent value into the optimized holdoff time exponent variable, decrements the target holdoff time exponent value by one, and starts the second iteration. During the second iteration, it finds that node 2 can win TxOpp 9. However, because the calculated TxOpp (9) is still larger than or equal to the target TxOpp, it enters the third iteration to probe further optimization.

During the third iteration, the algorithm finds that the calculated TxOpp is 5, which now is less than the target transmission opportunity. Thus, it stops this iterative procedure and returns the value stored in the optimized holdoff time exponent variable as its output. (Note: This value is the target holdoff time exponent value calculated in the previous iteration.) As can be seen in Fig. 3.3(b), upon performing the MEA with the optimized holdoff time exponent value (2 in this example), node 2 will win TxOpp 9, which is the optimal transmission timing to transmit its confirm IE in this example.



Request IE
Grant IE
Optimal Confirm IE Transmission Timing (iteration 2)
Possible Confirm IE Transmission Timing (iteration 1)

TxOpp	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Expected Winner	1	2	3	4	2	5	6	8	2	7	1	3	5	6	7	8	4	1	3	5	6	2	4	1	3	5

(a)

Iteration	Target TxOpp	Diff	Target Holdoff Time Exp	Calculated Holdoff Time	Starting TxOpp	Calculated TxOpp	Optimized Holdoff Time Exp
1	9	7	3	8	10	22	3
2	9	7	2	4	6	9	2
3	9	7	1	2	4	5	2

(b)

Figure 3.3: An example illustrating the operation of the dynamic approach

3.1.3 Effect of Holdoff Time Base Value

As introduced earlier, the IEEE 802.16 mesh CDS mode regulates that every node should set the holdoff time base value to 4. With this regulation, radios compliant to the 802.16 mesh CDS mode co-cooperate using this fixed holdoff time base value. Our proposed dynamic holdoff time scheme, however, may require changing the holdoff time base value to operate. For this reason, a mechanism is required to notify network nodes of changes to the holdoff time base value. In this section, we explain the problems that may occur if this system parameter is dynamically changed. In Section 3.1.4, we will describe several mechanisms that can be used to change this parameter without causing problems.

The holdoff time base value contributes a constant time amount (2^{base}) to the holdoff time. Setting the holdoff time base value to 4 means that each node should suspend its contention for at least 2^4 consecutive TxOpps after it has won one. This lower bound limits the smallest holdoff time value that can be assigned to nodes in the proposed dynamic holdoff time scheme. The proposed dynamic holdoff time scheme, therefore, cannot achieve its optimal performances, if the holdoff time base value is not reduced to 0. For this reason, in our proposed dynamic holdoff time scheme the holdoff time base value of the network is set to zero to provided the largest flexibility of control message scheduling.

Instead of using a lengthy field, the standard uses two shorter fixed-length fields, *exp* and *Mx*, to represent a TxOpp number. The relationship between these two fields and a TxOpp number has been given as follows:

$$2^{exp} \ * \ Mx < \text{next\_TxOpp} <= \ 2^{exp} \ * \ (Mx+1), \tag{3.3}$$

where next_TxOpp denotes a node's next TxOpp number.

On receiving a control message (such as an MSH-NCFG message or an MSH-DSCH message), a node should use the received *exp* and *Mx* fields to derive the next transmission interval of the transmitting node using Eq. (3.3). Fig. 3.4 shows the relationship between the holdoff time and the transmission interval (Tx Interval). The holdoff time comprises the Tx interval (denoted as α in the figure) and the ineligible interval (denoted as β in the figure). The Tx interval represents the duration in which a node may contend for one TxOpp. On the other hand, the ineligible interval is the duration in which the node is not allowed to contend for any TxOpp.

Based on Eq. (3.3), the length of the Tx interval is fixed to 2^{exp} because a node's next TxOpp number is within a fixed-length interval ranging from $(2^{exp} * Mx + 1)$ to $2^{exp} * (Mx + 1)$. Consequently, the length of the ineligible interval is ($2^{exp+base}$ - 2^{exp}). If the base value is set to 0, the holdoff time and the Tx interval of each node will exactly overlap, causing the length of the ineligible interval to be zero. This means that, after winning a TxOpp, a node will contend for another TxOpp immediately. (This also means that a node will contend for TxOpps all the time.) Thus, a node should consider that all nodes in its two-hop neighborhood will contend for each TxOpp with itself. On the other hand, if the holdoff time base value is larger than 0, a node's holdoff time will be larger than its Tx interval. In this condition, the contention time experienced by a node can be reduced due to a decreased number of contending nodes.

The choice of the holdoff time base value depends on the needs of a holdoff time scheme. For a static holdoff time scheme, using a positive holdoff time base value can reduce the contention time of each node. For a dynamic holdoff time scheme, however, the holdoff time base value must be zero for two reasons: First, if a positive holdoff time base value is used, the lower bound of the holdoff time value that can be assigned to nodes will be limited. Thus, to give the dynamic holdoff time scheme the largest freedom to set nodes' holdoff times, the holdoff time base value should preferably be set to zero at all

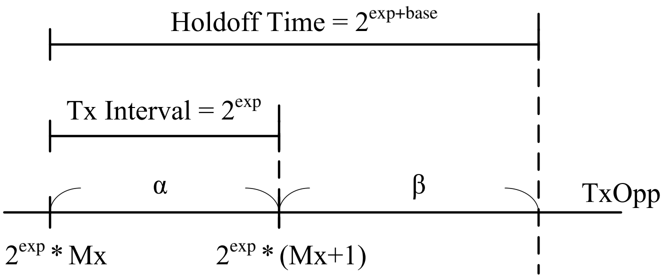


Figure 3.4: The relationship between the holdoff time and the Tx interval

time. Second, if the holdoff time base value is allowed to change during the operation of a network, after a node's holdoff time has just been changed (due to the change of the holdoff time base value), MSH-DSCH control messages may collide. The reason for this phenomenon is explained below.

Fig. 3.5 and Fig. 3.6 show two cases after a node's holdoff time value has just been changed. The former shows an example that changing the holdoff time value results in no message collisions while the latter shows an opposite example. Suppose that node A has changed its holdoff time value and broadcast the new holdoff time exponent value. In Fig. 3.5, node B is the next one to transmit an MSH-DSCH message. In this case, node C will be notified of this change by node B's MSH-DSCH message in time. Thus, node C will not schedule its MSH-DSCH message transmission to collide with node A's MSH-DSCH message transmission.

In contrast, in Fig. 3.6 node C has scheduled an MSH-DSCH message transmission before node B can notify it of node A's new holdoff time value. In this condition, node C's MSH-DSCH message transmission may collide with node A's MSH-DSCH message transmission because node A's ineligible interval viewed by node C now becomes out of date. Figures 3.7, 3.8, and 3.9 illustrate three cases that can cause this problem.

In these figures, HTa denotes the holdoff time of node A viewed by node C (may be out of date) and HTa$'$ denotes the holdoff time of node A viewed by node A itself (always up to date). The symbol γ denotes the vulnerable interval that results from node A's changing its holdoff time and during which the MSH-DSCH messages of nodes A and C may collide. Suppose that node C's transmission was scheduled within node A's original ineligible interval β. Fig. 3.7 depicts a case that node A has just decreased its holdoff time base value and therefore its ineligible interval is just shortened from β to β'. This operation generates the γ vulnerable interval because node C does not know that node A

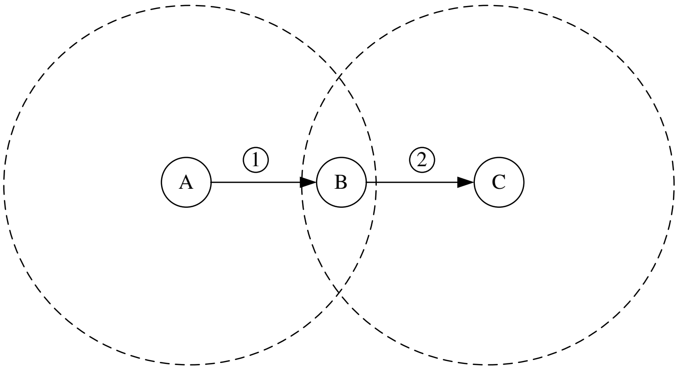


Figure 3.5: An example showing that control messages will not collide after node A changes its holdoff time value

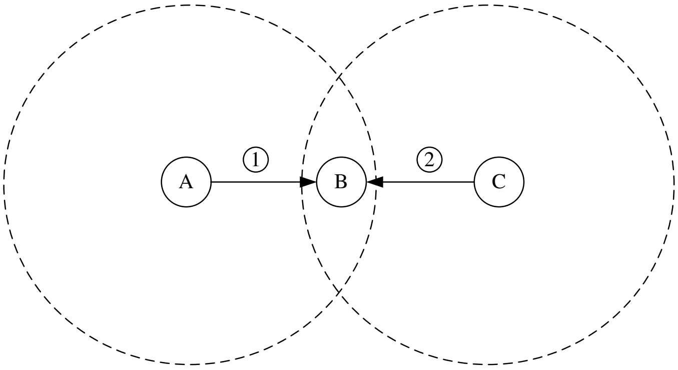


Figure 3.6: An example showing that control messages will collide after node A changes its holdoff time value

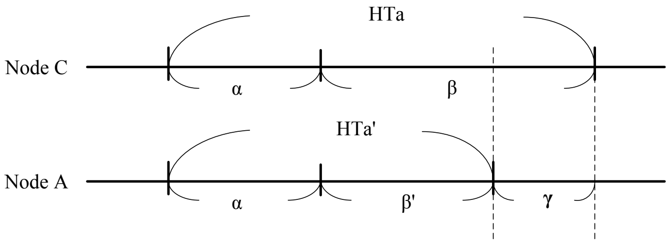


Figure 3.7: A case that node A has just decreased its holdoff time base value

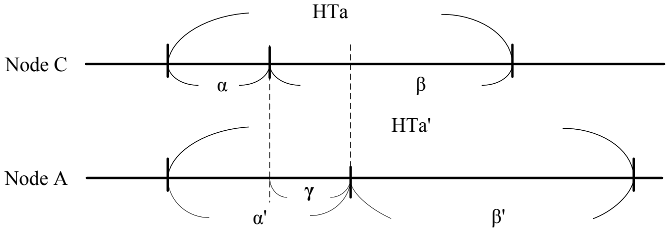


Figure 3.8: A case that node A has just increased its holdoff time exponent value

now will contend for TxOpps in the γ interval.

Fig. 3.8 depicts a case that node A has just increased its holdoff time exponent value and therefore both its Tx Interval and ineligible interval are lengthened. In this condition, node A's ineligible interval will shift on the time axis, generating the vulnerable interval shown in Fig. 3.8. In contrast, Fig. 3.9 depicts a case that node A has just decreased its holdoff time exponent value and therefore its Tx Interval and ineligible interval are shortened, resulting in the shift of node A's ineligible interval on the time axis. This shift generates the vulnerable interval shown in Fig. 3.9.

To prevent the collision problem from occurring, an additional mechanism to advertise nodes' changes to their holdoff time base and exponent values in time is needed. For instance, in the case given in Fig. 3.6, after changing the holdoff time value, node A should defer its contention for TxOpps until its original ineligible interval has elapsed. Such a mechanism, however, may increase the implementation complexity of a proposed

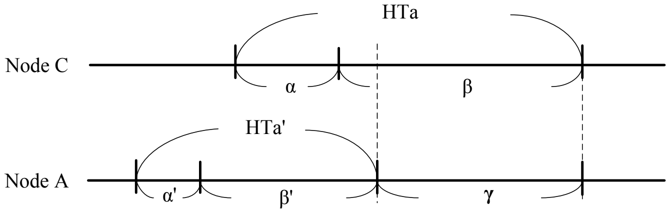


Figure 3.9: A case that node A has just decreased its holdoff time exponent value

dynamic holdoff time scheme and decrease its scheduling performances. To totally avoid the collision problem without wasting much network bandwidth, our proposed dynamic holdoff time scheme uses 0 as the holdoff time base value for all network nodes at all time.

Fixing the holdoff time base value to zero effectively eliminates every node's ineligible interval (i.e., the length of each node's ineligible interval now becomes zero.), resulting in each network node considering that it should always contend for TxOpps with all other nodes in its two-hop neighborhood. (These two-hop neighborhood nodes are considered to be always eligible to contend for TxOpps.) In this condition, if a node intends to win a TxOpp, it should win over all of its two-hop neighborhood nodes. This means that, for each node, the node list used as the input of MEA will always comprise its two-hop neighborhood nodes, despite the dynamic changes of the holdoff time exponent values of its neighboring nodes. Thus, packet collisions due to dynamic changes of the holdoff time exponent values can be avoided under the zero holdoff time base value condition.

3.1.4 Notification of Holdoff Time Base Value Change

Here, we propose a practical protocol that can notify new SS nodes of the holdoff time base value used in a network. A BS node can use this protocol to check whether a new SS node can operate using a holdoff time base value other than 4. If an SS node cannot do so, the BS node should reject the network registration request from this SS node because this SS node cannot work well with other SS nodes in the network.

Our proposed protocol exploits the reserved *"vendor-specific information"* field to help SS nodes know the holdoff time base value used in the network. This field is defined in the standard for the registration procedure to exchange additional information not specified in the standard. The signaling protocol is described as follows: An SS node first adds a holdoff time base query message (carried by the *"vendor-specific information"* field) into the registration request (REG-REQ) message, which is destined to the BS node.

On receiving this REG-REQ message, the BS node replies the SS node with a registration response (REG-RSP) message, which contains the holdoff time base value used in this network (also carried by the *"vendor-specific information"* field). If the BS node does not find the holdoff time base query message in the SS node's REG-REQ message, it should reject this SS node's registration request because this SS node may not be able to change its holdoff time base value.

There are three ways to reject a registration request. The first one is simply to ignore the REG-REQ message if the BS node decides to reject it. The second way is to utilize the "de/re-register command" (DREG-CMD) message, defined in the standard. The DREG-CMD message can be used to notify the SS node of the rejection action. The last way is to return a REG-RSP message with the response code set to 1, indicating that this registration request cannot be accepted because the SS node may not be capable of changing its holdoff time base value.

3.2 Proposed Scheme for Networks using Single-switched-beam Antennas

The effects of MEA on the performances of the 802.16(d) mesh CDS mode have been extensively studied [2][3][4][5][6]; however, most of the prior work were based on omnidirectional radios. In the literature, rare work studied the performances and challenges of the 802.16(d) mesh CDS-mode network when it uses directional antennas. Because the 802.16(d) mesh CDS mode assumes that MEA operates using omnidirectional radios, MEA encounters several operational problems when operating over directional radios.

Recent antenna technologies for directional message transmission/reception can be categorized into three classes. The first class is the switched-beam antenna, which uses pre-defined antenna gain patterns and can point to several pre-determined directions. Switched-beam antennas can operate optimally in environments without the presence of the multi-path effect (such as in the free-space environment). Due to lacking the nulling capability, however, they cannot achieve the optimal concurrent transmission schedules in multi-path-prone environments, as compared with the other two directional antenna classes.

The second antenna class is the adaptive array antenna, which is capable of forming arbitrary beams to point to arbitrary directions. By forming "nulls" to directions in which a transmitter does not intend to disseminate its signal or from which a receiver does not intend to sense the signal, adaptive array antennas can effectively reduce the multi-path effects. The last antenna class is the Multiple Input Multiple Output (MIMO) array antenna, which employs multiple antenna elements on both the transmitter and receiver ends. MIMO is well-known for its three capabilities: precoding, spatial multiplexing, and diversity coding, which can more effectively increase network capacity.

Although adaptive array antennas and MIMO antennas can be superior to switched-beam antennas in achieved network capacity and signal quality, the cost and complexity of their designs and implementations are much higher than those of switched-beam antennas. Due to the less design and implementation complexity, switched-beam antennas can be made with less form factor and at a lower cost; they therefore provide a cost-effective solution for wireless mesh networks using directional radios and more suitable for constructing emergent and tactic wireless mesh networks.

In this section, we reviewed the design of the 802.16 mesh CDS mode, identified the problems that result from using single-switched-beam antennas in this network, and proposed a scheme to solve these problems. The proposed scheme can operate using only directional transmissions/receptions. (Most of the previous proposals for directional antenna networks have to use omnidirectional transmissions/receptions in some phases of network operation.) We conducted proof-of-concept simulations to evaluate the network capacity increased by using single-switched-beam antennas in this network. In addition, we also evaluated the performances of TCP (Transport-layer Control Protocol) using a real-life TCP implementation in such networks. TCP is a well-known transport-layer protocol widely used in current network applications (such as FTP, HTTP, etc.) and is sensitive to network congestion and end-to-end packet delay jitters. Due to the unique protocol design of the 802.16 mesh CDS mode, how TCP performs under this network with single-switched-beam antennas is interesting and worth studying.

To the best of the authors' knowledge, our work is the first work that discusses how to enable the IEEE 802.16(d) mesh CDS-mode network to operate with single-switched-beam antennas and evaluates the performances of this network with single-switched-beam antennas. Although there have been many prior works studying TDMA networks with directional radios [7][8][9][10][11][12][13][14][15][16][17], they differ from the IEEE 802.16(d) mesh CDS mode using single-switched-beam antennas in either control message scheduling or data scheduling. Thus, the issues and performances of the IEEE 802.16(d) mesh CDS mode employing such an antenna configuration is worth studying. Although the IEEE 802.16 mesh mode may not be maintained in the next-generation 802.16 standard family due to several reasons, e.g., its design complexity is higher than that of the traditional point-to-multipoint (PMP) mode and its current business potential is less than that of the

PMP mode, in the literature so far it is one of the most representative WMNs that have been well developed and studied and can be a good design reference for next-generation WMNs.

In this dissertation, we do not assume that a single-switched-beam antenna allows omnidirectional transmission and reception because the antenna gain of such an antenna in the directional mode and that in the omnidirectional mode may greatly vary. Without proper transmission power control, the connectivity among nodes in the directional mode and that in the omnidirectional mode may be inconsistent and hinder network operation. To simplify the scheduling complexity, our proposed scheme operates only with pure directional transmission and reception. As a result, several protocol issues will arise due to such a harsh constraint. For example, in the IEEE 802.16(d) mesh CDS mode each node maintains its two-hop neighborhood to avoid the hidden-terminal problem when transmitting its control messages. The definition of such a two-hop neighborhood is based on the use of omnidirectional antennas and thus has an important property: *if node A is in node B's two-hop neighborhood, then node B is also in node A's two-hop neighborhood.* It is this property ensuring that the MEA used in each node generates collision-free TxOpp scheduling because node A and node B cannot both win the same TxOpp in their respective two-hop neighborhoods. However, when only pure directional transmission and reception are allowed (e.g., when using single-switched-beam antennas), the above property no longer holds at all time. This makes receiving control messages from other nodes non-trivial. In this condition, network operation and network initialization encounter several issues that need to be solved. In the following, we explain the problems of the IEEE 802.16(d) mesh CDS mode, when it uses single-switched-beam antennas to operate and initialize the network, and present our solutions to these problems.

3.2.1 Problem 1: Imprecise Representation for TxOpps in Control Messages

In the 802.16(d) mesh CDS mode, to save the bandwidth consumed by control messages, the next TxOpp number of a node carried in an MSH-DSCH message and an MSH-NCFG message is represented by a 5-bit Mx field and a 3-bit Exp field [1], rather than a single long field. Using this representation scheme, a TxOpp number is represented using the following formula:

$$2^{exp} * \mathrm{Mx} < \mathrm{TxOpp\ number} \leq 2^{exp} * (\mathrm{Mx} + 1), \tag{3.4}$$

where $0 \leq Mx \leq 30,\ 0 \leq exp \leq 7$. The interval between $(2^{exp} * \mathrm{Mx},\ 2^{exp} * (\mathrm{Mx}{+}1)]$ is called the next transmission interval of a control message in the standard.

It is known that using MEA no two nodes in the same two-hop neighborhood will use the same TxOpp to transmit messages. However, the transmission intervals of two nodes in the same two-hop neighborhood may overlay with each other. For example, consider two nodes A and B are in node C's two-hop neighborhood. Suppose that the current TxOpp is 0 and nodes A and B choose TxOpps 33 and 36 as their next MSH-DSCH TxOpps, respectively. In this condition, both nodes A and B may use $(2^5 * 1,\ 2^5 * 2]$ (base=4, exp=1, Mx=1) as their next transmission intervals and notify node C of these settings.

The overlapping of two neighboring nodes' transmission intervals does not hinder the operation of the 802.16(d) mesh CDS mode, when omnidirectional radios are used, because each node can listen incoming messages omnidirectionally, when it need not transmit control messages. In this condition, a node will not miss any control messages broadcast from neighboring nodes as long as it is not in the transmission state. However, when only directional reception is allowed, a node cannot determine to which direction its antenna should point because this imprecise representation cannot provide sufficient information for a node to know which node will transmit a message on a specific TxOpp among those nodes whose next transmission intervals are overlapped.

In addition, even though the transmission intervals of neighboring nodes do not overlap with each other, this imprecise representation scheme still reduces the flexibility of scheduling control message transmissions. Consider nodes A and B that are neighboring to each other. Using this TxOpp representation scheme, on receiving an MSH-DSCH message from node B, node A cannot know the exact next TxOpp number won by node B. Instead, it can only derive an interval of 2^{exp} TxOpps in length during which node B will broadcast its next MSH-DSCH message. Since node A cannot know the exact next TxOpp that node B wins, it has to point its antenna towards node B during the whole interval to successfully receive the next MSH-DSCH message broadcast from node B. However, if the exact next TxOpp won by node B can be known, node A can exchange control messages with some other nodes in this long interval to reduce the latency of

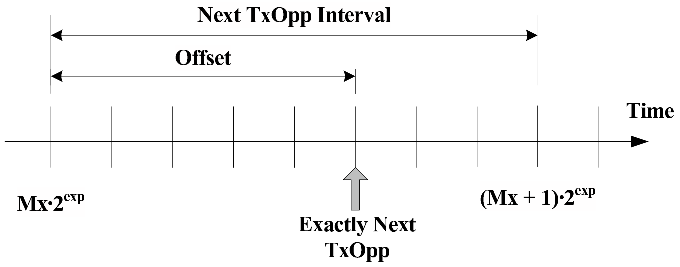


Figure 3.10: The auxiliary offset field for precisely representing a TxOpp number

updating network management information and scheduling data packet transmissions.

From this observation, to ensure that an 802.16(d) mesh CDS-mode network can obtain performance gains when using single-switched-beam antennas, a control message scheduling scheme has to control nodes' antenna directions in a per-TxOpp manner. To this end, our proposed scheme introduces a new *offset* field into the MSH-NCFG and MSH-DSCH message formats. As shown in Fig. 3.10, with the help of the newly-added *offset* field, a node now can use Eq. (3.5) to precisely derive the TxOpp numbers won by each of its neighboring nodes and thus know to which direction it should point the antenna on each TxOpp.

$$\text{TxOpp number} = 2^{exp} * \text{Mx} + \text{offset} \tag{3.5}$$

3.2.2 Problem 2: Control Message Transmissions using Pure Directional Transmission and Reception

The original MEA defined in [1] just schedules when to broadcast (omnidirectionally disseminate) control messages for nodes using omnidirectional antennas. Thus, it cannot be directly applied to networks using directional transmission and reception because, in such a network, a node using the original MEA cannot know to which direction it should point its antenna to transmit a control message on a given TxOpp. To solve this problem, we propose a distributed control message scheduling scheme called the Multiple Transmission Domain (MTD) Scheme to coordinate nodes' antenna pointings on each TxOpp. The MTD scheme uses a Transmission-domain-aware Mesh Election Algorithm (TMEA) and a Transmission-domain-aware Minislot Scheduling Algorithm (TMSA). The former is designed for a node to properly control when and in which direction it should trans-

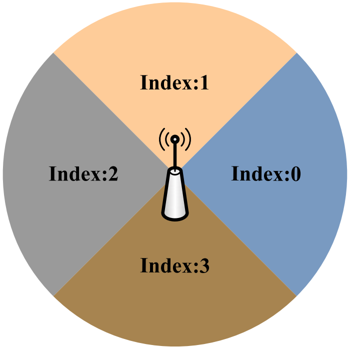


Figure 3.11: The four TDs used in the MTD scheme

mit a control message using a single-switched-beam antenna, while the latter is designed for a node to exploit the spatial-reuse advantage of single-switched-beam antennas to increase data transmission concurrency. In the following, we first explain the notion of a transmission domain (TD).

Transmission Domain and its Two-hop Neighborhood

In this section, we define the transmission domains (TDs) of a node and re-define the two-hop neighborhood of a node more exactly for directional-antenna networks. A TD of a node is defined as the set of nodes that are located in the coverage of a single switched beam and can simultaneously receive a message directionally transmitted by the node in that coverage. According to this definition, a node using an omnidirectional antenna has a single TD that includes all of its one-hop neighboring nodes in its 360-degree radio coverage. In contrast, a node using a single-switched-beam antenna has several TDs, each of which includes only the nodes in a specific beam coverage.

A node using a single-switched-beam antenna has $\frac{2\pi}{B}$ disjoint TDs to form a 360-degree radio coverage, where B denotes the antenna beamwidth in radians. Each TD is assigned a unique identifier i called the "*Transmission Domain Index (TDI).*" A TD i is composed of the nodes in the sector area between $B \cdot \left(i - \frac{1}{2}\right) \bmod 2\pi$ and $B \cdot \left(i + \frac{1}{2}\right) \bmod 2\pi$ in polar coordinates, $\forall\, i \in N,\ 0 \le i \le \frac{2\pi}{B}$. In this dissertation, B is set to $\frac{\pi}{2}$ radians but it can be changed to another value to better suit a given network topology. Therefore, each node has four disjoint TDs each comprising nodes in the areas of $\frac{\pi}{2} \cdot \left(i \pm \frac{1}{2}\right) \bmod 2\pi$ in polar coordinates, where $i \in N,\ 0 \le i \le 3$.

A node i using a single-switched-beam antenna maintains a two-hop neighborhood for each of its TDs. $nbr(i_j)$ denotes the set of nodes in node i's two-hop neighborhood associated with TD j. The definition of $nbr(i_j)$ is given as follows:

$$\begin{aligned} nbr(i_j) = \quad & \{i\} \cup \{k | k \text{ is in the radio coverage of TD } j\ \} \cup \\ & \{nbr_1(k) | k \text{ is in the radio coverage of TD } j\ \}, \end{aligned} \tag{3.6}$$

where the definition of $nbr_1(k)$ is given in Equation (2.2). The reason why we use $nbr_1(k)$ in the last term is that, a node k in TD j may point its antenna to any of its one-hop neighboring nodes (e.g., node l) to transmit/receive messages when node i intends to transmit a message to node k. For this reason, from the perspective of TxOpp scheduling, the message transmissions of all nodes in $nbr_1(k)$ may interfere with those of node i to node k. To take the interference of these nodes into consideration for scheduling collision-free TxOpps, the function $nbr_1()$ should be used in the definition of the two-hop neighborhood of node i's TD j.

Using multiple TDs to maintain multiple two-hop neighborhoods on a node using a single-switched-beam antenna is the fundamental idea of the proposed MTD scheme. The details of how the MTD scheme coordinates the TxOpp scheduling of different TDs are explained below in Section 3.2.2. For comparison, in the remainder of this paper, the 802.16(d) mesh CDS mode that operates over an omnidirectional-antenna network is referred to as the single-transmission-domain (STD) scheme.

Transmission-domain-aware Mesh Election Algorithm (TMEA)

As described previously, the MEA defined in [1] was originally designed for a node using omnidirectional radios (i.e., using a single TD). In the previous section, we have discussed why the original MEA cannot be directly used in a network using single-switched-beam antennas. One solution for such a network is to run multiple MEAs in each TD of a node. However, using this approach will result in several problems, which necessitates a revision of the original MEA. In the following, we first explain these problems and present a version of TMEA that uses a static holdoff time design. We then discuss its drawbacks and present a version of TMEA that exploits a dynamic holdoff time design to boost the performances of TMEA and the MTD network.

Consider an IEEE 802.16(d) mesh CDS-mode network using single-switched-beam antennas, where each node has four TDs. Following the design defined in [1], each node should execute four MEA instances (denoted as MEAIs), each maintaining a two-hop neighborhood for a specific TD and calculating the next TxOpp number for that TD.

For brevity, the MEAI for a TD k is denoted as "TD-k MEAI." Because the 802.16 mesh CDS mode [1] does not define how multiple MEAIs of the same node operate and coordinate with each other, these MEAIs independently perform their next TxOpp number calculations. If the next TxOpp numbers calculated by the different MEAIs running on a node turn out to be different, there is no conflict among these MEAIs. The node just needs to point its single-switched-beam antenna towards a TD to broadcast a control message, when its current TxOpp number advances to the calculated next TxOpp number for that TD.

However, without coordination, multiple MEAIs running on a node may choose the same TxOpp because they do not know whether the TxOpp that they choose has already been chosen by other MEAIs. In such a case, because at any given time a node can only broadcast a control message towards a TD, the network scheduling efficiency will decrease. For example, suppose that both TD-1 MEAI and TD-2 MEAI choose the same TxOpp to broadcast their messages and only TD-1 MEAI can be served by the node. Because TD-2 MEAI cannot be served and only one MEAI (i.e., this TD-2 MEAI) in its two-hop neighborhood can win this TxOpp, this TxOpp is wasted in this two-hop neighborhood as no MEAI of other node in this two-hop neighborhood can broadcast its message in this TxOpp. Such a scheduling conflict is called an "intra-node scheduling conflict" in this dissertation as it takes place among the TDs running on the same node.

To eliminate the intra-node scheduling conflicts, in this section we propose a version of TMEA that uses the static holdoff time design defined by the standard and exploits the original MEA for nodes' MEAIs to coordinate their scheduling decisions. For brevity, in the following context we call this version of TMEA that uses the static holdoff time design TMEA-S. The pseudo code of TMEA-S running on node i for TD j is shown in Algorithm 1 and explained below.

Using TMEA-S, the MEAI first sets the contention start TxOpp number to the current TxOpp number plus the holdoff time. (Recall that, in the 802.16(d) mesh CDS mode, each MEAI should suspend its contention for TxOpps for a holdoff time interval after it has won a TxOpp.) The contention start TxOpp is the first TxOpp for which this MEAI is allowed to contend. This MEAI then exploits the original MEA (denoted as MEA$_{ori}$) to calculate the next TxOpp number for its TD. The first parameter fed to MEA$_{ori}$ is the contention start TxOpp while the second parameter fed to MEA$_{ori}$ is L_{ij}, which is

Algorithm 1 TMEA-S

```
$L_{ij} := \{(k, L_{ij}(k)) | \forall$ TxOpp $k$, $L_{ij}(k)$ is the eligible node list of node $i$'s TD $j$,
 constructed based on the eligibility of all nodes in $\cup\, nbr(i_m), \forall$ TD $m \in$ node $i\}$
reference_start_txopp := current TxOpp
next_txopp := current TxOpp + holdoff_time
contention_start_txopp := next_txopp
while true do
    next_txopp $\leftarrow$ MEA$_{ori}$(contention_start_txopp, $L_{ij}$)
    if next_txopp has been used by other MEAIs on the same node then
        contention_start_txopp := next_txopp + 1
    else
        break
    end if
end while
proper_exp := floor($log_2$(next_txopp − reference_start_txopp))
proper_offset := (next_txopp − reference_start_txopp) − $2^{proper\_exp}$
return (next_txopp, proper_exp, proper_offset)
```

the set of eligible node lists for each TxOpp for the TD j of node i derived from the eligibility of all nodes in $\cup nbr(i_m), \forall$ TD $m \in$ node i. Note that the construction of L_{ij} should consider the eligibility of all nodes in $\cup nbr(i_m), \forall$ TD $m \in$ node i rather than the eligibility of nodes only in $nbr(i_j)$ to avoid the "inter-node scheduling conflict" problem. The reason will be explained soon below.

This MEAI then checks whether the calculated TxOpp number has been won by another MEAI. (Since all of the MEAIs of a node are run on the same node, this information sharing can be easily achieved.) If not, the MEAI calculates proper exp and offset values for this TxOpp number and then returns the 3-tuple (next_txopp, proper_exp, proper_offset) as its output. Otherwise, it sets the contention start TxOpp number to the TxOpp number that it just calculated plus 1 and then performs MEA$_{ori}$ again. Such a procedure is repeated until the MEAI wins a TxOpp number that has not been won by other MEAIs. Note that, for brevity, the calculation for the proper_exp and proper_offset values shown here does not consider the sequence number wrapping problem of tmp_txopp, which has been properly solved in our implementation.

Fig. 3.12 shows an example case illustrating the operation of the TMEA-S for TD 1. In this example, the TD-1 MEAI is scheduling its next control message transmission. At the first iteration, the TD-1 MEAI chooses a TxOpp number that has been won by the TD-3 MEAI. Thus, the TD-1 MEAI sets its contention start TxOpp number to this TxOpp number plus 1 and starts its second iteration. Again, at the second iteration the TxOpp number that it chooses has been won by the TD-2 MEAI. Thus, it has to repeat the above process again. Eventually, at the third iteration the TD-1 MEAI chooses a

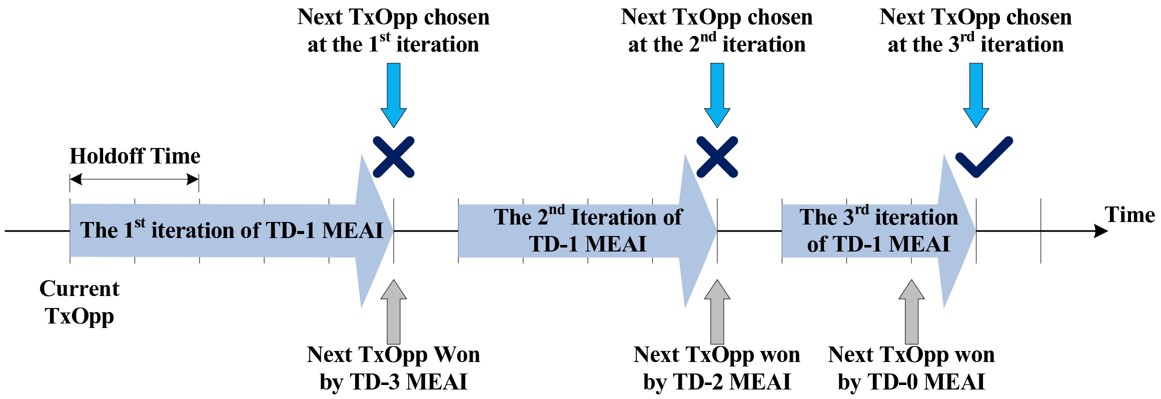

Next TxOpp chosen at the 1st iteration
Next TxOpp chosen at the 2nd iteration
Next TxOpp chosen at the 3rd iteration
Holdoff Time
The 1st iteration of TD-1 MEAI
The 2nd Iteration of TD-1 MEAI
The 3rd iteration of TD-1 MEAI
Time
Current TxOpp
Next TxOpp Won by TD-3 MEAI
Next TxOpp won by TD-2 MEAI
Next TxOpp won by TD-0 MEAI

Figure 3.12: An example case illustrating the operation of TMEA-S associated with TD 1

TxOpp number that has not been won by the MEAIs of other TDs and thus can return it as output. By using the TMEA-S, an MEAI can now safely determine its next TxOpp number without generating intra-node scheduling conflicts.

The reason why the TMEA-S of node i for its TD j should use an eligible node list derived from the holdoff times of nodes in $\cup nbr(i_m), \forall$ TD $m \in$ node i, rather than an eligible node list derived from the holdoff times of nodes in $nbr(i_j)$ is explained here. Consider a simple example network shown in Fig. 3.13. In this network, node A is located in node B's TD 2 and node B is located in node A's TD 0. Assume that the TD-0, TD-1, and TD-3 MEAIs of node A have chosen TxOpps 5, 2, and 3, on which to broadcast their MSH-DSCH messages to TD 0, TD 1, and TD 3, respectively, and node B has chosen TxOpp 8 on which to broadcast its MSH-DSCH messages to its TD 2. On the right of this figure, we show that, starting TxOpp 0, node B's TD-2 MEAI is performing TMEA-S to calculate its next TxOpp and finally TxOpp 8 was calculated as its winning TxOpp number.

Later on, when the TD-2 MEAI of node A calculates its next TxOpp on which to broadcast an MSH-DSCH message to its TD, if the TD-2 MEAI of node B is not included in its two-hop neighborhood (which is included only in $nbr(A_0)$), it may also choose TxOpp 8 as its next TxOpp. This will cause a scheduling conflict on TxOpp 8. When node B sends an MSH-DSCH message to node A on TxOpp 8, node A will not be able to receive its message because node A will point its antenna to its TD 2 on TxOpp 8. Because such a scheduling conflict is caused by the MEAIs belonging to different nodes, it is called an "inter-node scheduling conflict" in this dissertation.

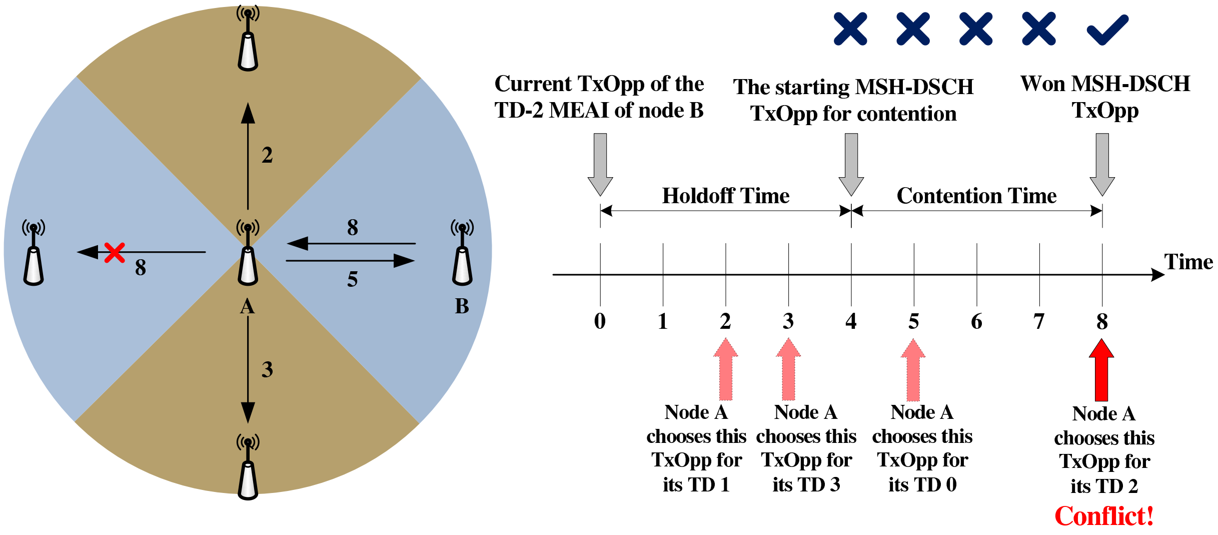

2
8
8
5
A
B
3
Current TxOpp of the TD-2 MEAI of node B
The starting MSH-DSCH TxOpp for contention
Won MSH-DSCH TxOpp
Holdoff Time
Contention Time
Time
0
1
2
3
4
5
6
7
8
Node A chooses this TxOpp for its TD 1
Node A chooses this TxOpp for its TD 3
Node A chooses this TxOpp for its TD 0
Node A chooses this TxOpp for its TD 2
Conflict!

Figure 3.13: An example case illustrating the inter-node scheduling conflict problem

To avoid such scheduling conflicts, TMEA-S requires each node to notify its one-hop neighboring nodes of the next TxOpp numbers of all its own MEAIs and those of all its one-hop neighboring nodes' MEAIs (learned from received MSH-DSCH messages transmitted by its one-hop neighboring nodes) using MSH-DSCH messages. With the complete next TxOpp information of all neighboring MEAIs, TMEA-S can build an eligible node list based on $\cup nbr(i_m), \forall$ TD $m \in$ node i to avoid inter-node scheduling conflicts.

For instance, for the example shown in Fig. 3.13, using this design the TD-2 MEAI of node A will include node B in its eligible node list for TxOpp 8 according to the next TxOpp information of node B's last MSH-DSCH message. Because there can be only one winner in any two-hop neighborhood, the TD-2 MEAI of node A will not choose the same TxOpp 8 as the TD-2 MEAI of node B did, resolving a potential inter-node scheduling conflict.

Note that, according to the MEA defined in the 802.16(d) mesh CDS mode, when the next TxOpp of a neighboring node in node i's two-hop neighborhood is unknown, node i should include this node into its own eligible node lists for subsequent TxOpps, until node i receives the next TxOpp information of this neighboring node. We expand this node-based conservative eligibility determination rule into a TD-based form which is given as follows: *"When the next TxOpp of a neighboring node's MEAI in node i's two-hop neighborhood is unknown, each MEAI of node i should include this MEAI into its own eligible node lists for subsequent TxOpps, until node i receives the next TxOpp information of this MEAI."* Using this conservative eligibility determination rule avoids another type of the inter-node scheduling conflicts resulting from unknown and obsolete next TxOpp information of neighboring nodes. For example, for node i, when the next MSH-DSCH message of node j that are two-hop away from it cannot be received before node i schedules its own next MSH-DSCH TxOpp, using this conservative eligibility determination rule each MEAI of node i will include node j into its own eligible node list and conservatively consider that an MEAI of node j will contend for each subsequent TxOpp. Thus, when an MEAI of node i wins a TxOpp, it will ensure that all node j's MEAIs will not win the same TxOpp, which prevents inter-node scheduling conflicts among two-hop neighboring nodes from occurring.

3.2.3 TMEA using a Dynamic Holdoff Time Design

To solve the inter-node scheduling conflicts, each TMEA-S of the same node's TD has to use an eligible node list derived from all $nbr(i_m)$, $\forall\ m$ on the same node, and the TD-based conservative eligibility determination rule. However, these two designs of TMEA-S may make the MEAIs of the same node derive nearly the same eligible node list. In this condition, the diversity of the inputs of MEA$_{ori}$, which chooses a winning node of a given TxOpp using a pseudo-random hash function, is very limited. Thus, the winning TxOpp of each MEAI on the same node is likely to be the same at the first k iterations (which continuously results in intra-node scheduling conflicts), where k is the number of TDs on a node.

This is because in this condition each MEAI's eligible node list for the same TxOpp may have many common nodes (worse yet, their eligible node lists may be the same for many TxOpps due to the conservative TD-based eligibility determination rule), MEAIs on the same node may, therefore, find that in MEA$_{ori}$ the winning nodes on each TxOpp calculated by them (generating the largest hash values) are the same. As a result, they will win the same next TxOpp when executing MEA$_{ori}$, which generates many intra-node scheduling conflicts, until some of them increase their contention_start_txopp values to enough large values. We call this problem the "continuous intra-node scheduling conflict" problem, which will increase the interval of each MEAI's transmission cycle and therefore increase the required time to negotiate a data transmission. In this condition, the utilization of the link bandwidth will be decreased due to this problem.

To prevent this problem from occurring, we propose a version of TMEA that uses a

Table 3.1: The computation cost of TMEA-S and TMEA-D

	Contention Time	**Contending Node List Size**	**Computation Complexity**
TMEA-S	$\frac{NTD(NTD+1)}{2} * nbr_2(x)$	$nbr_2(x)$	$O(NTD^2 * nbr_2(x))$
TMEA-D	$NTD * nbr_2(x)$	$nbr_2(x)$	$O(NTD * nbr_2(x))$

dynamic holdoff time design (denoted as TMEA-D) to increase the diversity of the inputs fed into MEA$_{ori}$. TMEA-D has four advantages: First, it provides more scheduling flexibility for MEAIs to avoid continuous intra-node scheduling conflicts by allowing each MEAI on the same node uses distinct holdoff time values. Second, TMEA-D can generate fair TxOpp scheduling for MEAIs on the same node. Thirdly, TMEA-D can assign active MEAIs (those that have data to send) smaller holdoff time exponent values and idle MEAIs larger holdoff time exponent values. Thus, using TMEA-D the time required for a transmitting node to handshake a minislot allocation can be reduced, which allows nodes to more efficiently utilize link bandwidth. Finally, because TMEA-D can avoid continuous intra-node scheduling conflicts, it can greatly reduce the number of computation operations, as compared with TMEA-S.

We show the computation cost of these two schemes in Tab. 3.1, where NTD denotes the number of TDs used by a node. As can be seen, the computation complexity of TMEA-D is linear to NTD while that of TMEA-S is proportional to NTD^2, which shows the TMEA-D is more computation-efficient than TMEA-S.

The operation of TMEA-D is shown in Algorithm 2. In the initialization phase, an MEAI first sets its smallest_tmp_txopp and reference_start_txopp to the current TxOpp and constructs its L_{ij}. It then empties the set S$_{calc}$, which is used to store the TxOpp numbers that are chosen by this MEAI when executing Algorithm 2 but have been used by other MEAIs. If the node has data to send in the TD, the MEAI of this TD randomly chooses a holdoff time exponent value exp from S$_{active}$. S$_{active}$ is composed of smaller exp values from 0 to the number of TDs where this node has data to send. We call such a TD an active TD and denote the number of active TDs as Num_of_ActTDs in Algorithm 2. The chosen exp value is stored in the tmp_exp variable and used to derive the contention_start_txopp, which is the reference_start_txopp plus 2^{tmp_exp}. Then, an MEAI calculates the next TxOpp that it can win (stored in tmp_txopp) using the same hash function used by MEA$_{ori}$ with the chosen contention_start_txopp value and L$_{ij}$.

If the chosen next TxOpp is not used by other MEAIs on the same node, it first

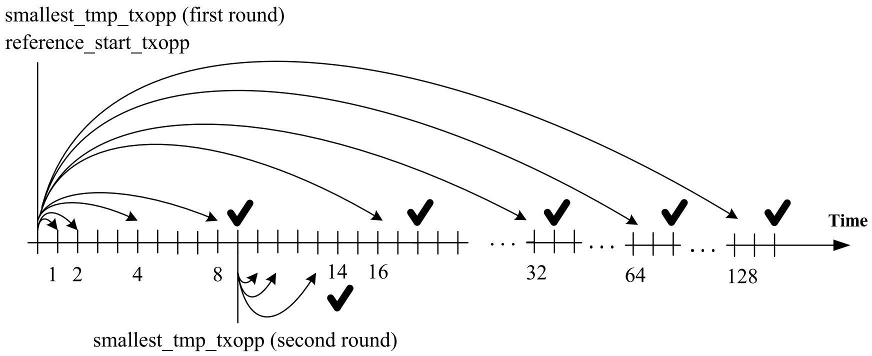

smallest_tmp_txopp (first round)
reference_start_txopp
Time
1
2
4
8
14
16
32
64
128
smallest_tmp_txopp (second round)

Figure 3.14: An example of the iterations of TMEA-D

calculates proper *exp* and *offset* values for this chosen TxOpp number and then return the 3-tuple (tmp_txopp, proper_exp, proper_offset) as its output. Otherwise, it adds the chosen next TxOpp number into S_{calc}, removes the used tmp_exp value from S_{active}, and repeats the above process until it wins a TxOpp that has not been used by other MEAIs on the same node. (Note that, for brevity, the calculation for the proper_exp proper_offset values shown here does not show the details for processing the tmp_txopp value wrapping problem, which has been properly solved in our implementation.)

In case that S_{active} becomes empty, it means that this MEAI cannot find a tmp_txopp value that has not been won by other MEAIs on the same node using the contention_start_txopp values derived from the current smallest_tmp_txopp value and the *tmp_exp* values in S_{active}. To address this problem, the MEAI advances its smallest_tmp_txopp to the minimum TxOpp stored in the S_{calc} (which is the smallest TxOpp that it wins in the iterations of the while loop) and then re-performs the above process. By doing so, this MEAI can have a chance to win a TxOpp that has not been won by other MEAIs belonging to the same node. Such an iterative process repeats until the MEAI finally wins an unused TxOpp that has not been chosen by other MEAIs on the same node.

Fig. 3.14 illustrates an example of how TMEA-D solves the above problem. Suppose that a node i has six MEAIs and its TD-1 MEAI is calculating the next MSH-DSCH TxOpp. First, the TD-1 MEAI sets its smallest_tmp_txopp to 0 and then performs MEA_{ori} to find its next TxOpp number. In the first round, it finds that the TxOpp numbers that node i can win have been used by other MEAIs on node i (TxOpps 9, 18, 33, 66, and 130). As a result, it advances its smallest_tmp_txopp to 9 and restarts the next TxOpp finding process. In this round, it successfully wins TxOpp 14 that has not been won by other MEAIs of node i.

Algorithm 2 TMEA-D

```
1:  Num_of_ActTDs := the number of TDs on the same node where there is data to send
2:  $L_{ij} := \{(k, L_{ij}(k))| \forall$ TxOpp $k$, $L_{ij}(k)$ is the eligible node list of node $i$'s TD $j$,
3:   constructed based on the eligibility of all nodes in $\cup\, nbr(i_m), \forall$ TD $m \in$ node $i\}$
4:  smallest_tmp_txopp := current TxOpp
5:  reference_start_txopp := current TxOpp
6:  $S_{calc} := \emptyset$
7:  $S_{active} := \{0, 1, 2, 3, ....$ , min( Num_of_ActTDs -1, max_exp)$\}$
8:  $S_{calc} \leftarrow \emptyset$
9:  if there is data to send in my TD then
10:     found_flag := false
11:     while $S_{active} \neq \emptyset$ do
12:         tmp_exp := random($S_{active}$)
13:         contention_start_txopp := smallest_tmp_txopp + $2^{tmp\_exp}$
14:         tmp_txopp $\leftarrow$ $MEA_{ori}$(contention_start_txopp, $L_{ij}$)
15:         if tmp_txopp has used by other MEAIs on the same node then
16:             $S_{calc} \leftarrow S_{calc} \cup \{$tmp_txopp$\}$
17:             $S_{active} \leftarrow S_{active} - \{$tmp_exp$\}$
18:         else
19:             found_flag $\leftarrow$ true
20:             break
21:         end if
22:     end while
23:     if found_flag = false then
24:         smallest_tmp_txopp $\leftarrow$ min($S_{calc}$)
25:         goto line 7
26:     end if
27: else
28:     found_flag := false
29:     if min( Num_of_ActTDs, max_exp) $\leq \lceil \frac{max\_exp}{2} \rceil$ then
30:         starting_exp := $\lceil (\frac{max\_exp}{2}) \rceil$
31:     else
32:         starting_exp := min( Num_of_ActTDs, max_exp)
33:     end if
34:     for tmp_exp $\leftarrow$ starting_exp to max_exp do
35:         contention_start_txopp := smallest_tmp_txopp + $2^{tmp\_exp}$
36:         tmp_txopp $\leftarrow$ $MEA_{ori}$(contention_start_txopp, $L_{ij}$)
37:         if tmp_txopp has been used by other MEAIs on the same node then
38:             $S_{calc} \leftarrow S_{calc} \cup \{$tmp_txopp$\}$
39:             contention_start_txopp $\leftarrow$ contention_start_txopp + 1
40:         else
41:             found_flag $\leftarrow$ true
42:             break
43:         end if
44:     end for
45:     if found_flag = false then
46:         smallest_tmp_txopp $\leftarrow$ min($S_{calc}$)
47:         goto line 8
48:     end if
49: end if
50: proper_exp := floor($log_2$(tmp_txopp − reference_start_txopp))
51: proper_offset := (tmp_txopp − reference_start_txopp) − $2^{proper\_exp}$
52: return (tmp_txopp, proper_exp, proper_offset)
```

The reason why TxOpp 14 cannot be derived in the first round but can be derived in the second round is explained here. TxOpp 14 is after TxOpp 9, which is found when the exponent value is advanced to 3 (i.e., the holdoff time becomes $2^3 = 8$), and before TxOpp 18, which is found when the exponent value is advanced to 4 (i.e., the holdoff time becomes $2^4 = 16$). According to the standard, when MEA$_{ori}$ wins TxOpp 9, it should immediately return that TxOpp as the output. Later on, when MEA$_{ori}$ finds that TxOpp 9 has been won by another MEAI and a new one should be found, it advances the exponent value from 3 to 4 and starts the searching from TxOpp 16=0+16 (where 0 is the value of smallest_tmp_txopp and 16 is the current holdoff time 2^4). However, this exponential holdoff time expansion causes TxOpp 14 to be skipped in the search during the first round. At the second round, because smallest_tmp_txopp is moved to TxOpp 9 and the exponent value starts over from 0 again, the search can start from TxOpp $10=9+2^0$ and eventually find TxOpp 14.

For the case where there is no data to send in the TD, the MEAI need not use a small next TxOpp to transmit its next control message. Thus, it first determines the starting holdoff time exponent value (stored as starting_exp) using the minimum between Num_of_ActTDs and max_exp[1]. If this value is below $\lceil \frac{\text{max\_exp}}{2} \rceil$ (i.e., 4), then MEAI adjusts it to 4. The rationale behind this design is that the number of active TDs of a node may dynamically fluctuate. Because the node has no data to send in this TD, the control message dissemination of this TD is not time-critical. Thus, TMEA-D prevents the MEAI of an idle TD from using small holdoff time exponent values, which are more valuable for MEAIs of active TDs to reduce the time required for negotiating minislot allocations.

After determining the starting_exp value, the MEAI of an idle TD iteratively finds its next TxOpp using holdoff time exponent values from start_exp to max_exp. The calculation in this iterative process is similar to that used by an MEAI of an active TD. The main difference is that an MEAI of an active TD chooses a smaller holdoff time exponent value from S$_{active}$ in a random manner while an MEAI of an idle TD chooses a larger holdoff time exponent value from [max(4, min(Num_of_ActTDs,max_exp)), max_exp] in an iterative manner. For brevity, we do not repeat the same explanation for these calculations here.

[1] The maximum holdoff time exponent value is defined as 7 in the standard.

As one sees, the MEAIs of active TDs on the same node share the useable smaller holdoff time values in a random manner. This design prevents some of them from monopolizing the smallest holdoff time values and thus ensures a fair sharing of these valuable small holdoff time values among them in a long term. Thus, the average times required by the MEAIs of a node's active TDs to negotiate minislot allocations (to transmit data) can be the same.

3.2.4 Problem 3: Network Initialization

In the 802.16(d) mesh CDS mode, a new node is required to transmit its MSH-NENT messages on NENT TxOpps for requesting to join the network in a contention-based manner, while a node that has been operational[2] should transmit its MSH-NCFG messages on NCFG TxOpps for network maintenance and its DSCH messages on MSH-DSCH TxOpps for minislot scheduling in a collision-free manner (with the aid of MEA/TMEA-S/TMEA-D). Initializing an 802.16(d) mesh CDS-mode network using single-switched-beam antennas encounters four problems: 1) how a new node can receive MSH-NCFG messages transmitted by neighboring operational nodes to synchronize its clock and MAC-layer framing with those of the network; 2) when a new node can transmit its MSH-NENT messages so that its chosen sponsoring operational node can receive them; 3) how an operational node can receive an MSH-NENT message transmitted from a neighboring new node; and 4) when an operational node can transmit its MSH-NCFG messages so that neighboring new nodes can have chances to receive them. We answer these questions and briefly explain our refined network entry process for a new node using a single-switched-beam antenna below.

First, each operational node is required to periodically transmit its MSH-NCFG messages to each of its TDs regardless of the inexistence of operational nodes in those TDs. That is, the MEAIs of all the TDs of an operational node should be activated at all time after the node has joined the network. With this design, a new node can have a chance to receive MSH-NCFG messages of its neighboring operational nodes. For a new node, it should first scan all of its TDs to detect whether any operational node exists. This scanning procedure is described here. When a new node boots, it should iteratively point

[2] Recall that a node being operational means that it has joined the network and started transmitting network-management messages.

its antenna to each of its TD to try receiving any incoming messages. When pointing its antenna to a TD, the new node should continuously listen to this TD for 2^8 NCFG TxOpps, unless it receives MSH-NCFG messages sent by the operational nodes in this TD. Because each operational node is allowed to use holdoff times ranged from 2^0 to 2^7, continuously listening to a TD for 2^8 NCFG TxOpps is sufficient for a new node to receive an operational node's MSH-NCFG messages, if any operational node exists in that TD. If the new node does not receive any messages in a TD, after listening for 2^8 NCFG TxOpps, the new node should point its antenna to next TD and continue its scanning process. A new node should repeat the above process until it has scanned all of its TDs and detected the existence of neighboring operational nodes.

According to [1], a new node cannot transmit its own MSH-NENT message until it receives the MSH-NCFG messages sent from the same node twice. Thus, after detecting the existence of an operational node, the new node can stay in the TD where the detected operation node resides to wait for its next MSH-NCFG message. If multiple operational nodes in different TDs are detected, the new node can first wait for the MSH-NCFG messages transmitted by the operational node with the highest SINR value.

After receiving MSH-NCFG messages from the same node twice, the new node can choose one of its neighboring operational node as its sponsoring node and start its network entry process. In the network entry process, the sponsoring node is responsible for allocating a temporary bidirectional minislot allocation to relay network-joining control messages transmitted by the base station and the new node. This temporary minislot allocation is required because a new node has not been operational and thus cannot negotiate minislot allocations with any operational nodes.

In the network entry process, the new node is required to transmit several network-joining and registration control message to the base station. These control messages should be relayed via its chosen sponsoring node and be first sent to the sponsoring node using MSH-NENT messages on NENT TxOpps. To achieve this goal, two problems need to be solved: One problem is when a new node can transmit its MSH-NENT messages on certain NENT TxOpps so that its chosen sponsoring node can receive them. The other problem is how an operational node can receive such network-joining control messages from an unknown (new) node on NENT TxOpps. We first answer the second problem here: An operational node is required to iteratively point its antenna to each of its TDs on NENT TxOpps. That is, on each NENT TxOpp, it points its antenna to a different TD. For example, a k-TD node has to point the antenna to its TD ($i \ mod \ k$) on NENT TxOpp i.

With the above arrangement, we describe the solution for the first problem here: For a new node, it first divides NENT TxOpps into several k-TxOpp groups. After choosing its sponsoring node, the new node should point its antenna to the TD where the sponsoring node resides on all NENT TxOpps. It then transmits its MSH-NENT messages that carry its network-joining control messages in a probabilistic manner. When the NENT TxOpp number advances to the boundary of a k-TxOpp group, a new node randomly chooses a fraction N_f between 0 and 1. If N_f is below a pre-determined value N_{th}, the new node is allowed to transmit its network-joining messages during this TxOpp group. In this condition, it should continuously transmit its MSH-NENT messages on each of the k TxOpps in this TxOpp group. If N_f is larger than or equal to N_{th}, the new node should keep silent during this TxOpp group. Because an operational node sequentially switches its antenna to each of its k TDs on k TxOpps and a new node stays at one of its TD for k TxOpps, their antennas can meet each other on a certain TxOpp.

If the chosen sponsoring node receives an MSH-NENT message from a new node on NENT TxOpps, it should first add the ID of this new node into the eligible node list of each of its TD and considers the control message schedulings of this new node for MSH-NCFG and MSH-DSCH messages are in unknown statuses. That is, it should assume that the new node will contend for every MSH-NCFG and MSH-DSCH TxOpp; thus, its MEAIs should always take the contention of this new node for NCFG and DSCH TxOpps into consideration. In addition, the sponsoring node should disseminate the unknown schedule information of this new node to its neighboring nodes using its MSH-NCFG and MSH-DSCH messages, such that its neighboring nodes can also take the contention of this new node in consideration. The reason behind this design is that, if the nodes in the new node's two-hop neighborhood do not consider its contention for NCFG and DSCH TxOpps in advance, after the new node becomes operational, the one-hop neighboring nodes of the new node will not point their antenna to receive the MSH-NCFG and MSH-DSCH messages transmitted by the new node and the two-hop neighboring nodes of the new node will not properly point their antennas to avoid inter-node scheduling conflicts.

The new node can know whether its chosen sponsoring node has received and accepted its sponsorship request by checking whether the next MSH-NCFG message transmitted by the chosen sponsoring node contains the network-entry-open IE for acknowledging its sponsorship request. If the new node does not find the corresponding network-entry-open IE in the sponsoring node's next MSH-NCFG message, it should halve the value of N_{th} and repeat the above process until the chosen sponsoring node responses it with a network-entry-open IE. The reason why a new node should decrease the value of N_{th} is to reduce the contention for NENT TxOpps, if multiple new nodes exist and simultaneously choose the same operational node as their sponsoring nodes. The lower bound of N_{th} is set to 0.1 in our simulations, which is sufficient for all new nodes to join the simulated network. On the other hand, if the new node finds a network-entry-open IE for another new node in the next MSH-NCFG message of the chosen sponsoring node, it should cease its MSH-NENT message transmission to avoid disturbing the ongoing network entry process of another new node, until it finds that the chosen sponsoring node transmits an MSH-NCFG message containing a network-entry-ack IE for that new node, which indicates that that new node has completed its network entry process.

If the new node detects a network-entry-open IE for itself in the chosen sponsoring node's MSH-NCFG message, it means that the sponsoring node has allocated a temporary bidirectional minislot allocation for the new node to relay its network-joining and registration control messages. The new node should then transmit a network-entry-ack IE to its sponsoring node using MSH-NENT messages for acknowledging its sponsoring node (using the same way for transmitting its MSH-NENT message). After this, it can transmit its capacity negotiation, authorization, and network registration control messages to the sponsoring node over the temporary minislots allocated by the sponsoring node for it. After receiving these control messages, the sponsoring node will in turn forward these messages towards the BS node in the network. Also, upon receiving control messages transmitted by the BS node and destined to the new node, the sponsoring node will forward them to the new node on the temporary minislots that it allocated for the new node.

After finishing exchanging necessary control messages with the BS node, the new node has joined the network and become operational. It should then transmit a network-entry-close IE to the sponsoring node using MSH-NENT messages on NENT TxOpps. The network-entry-close IE is used to notify the sponsoring node that the network-entry process of the new node that it sponsors has been completed and the temporary minislot allocation reserved for the new node is no longer needed. After receiving the network-entry-close IE, the sponsoring node first acknowledges the new node by sending it an MSH-NCFG message containing a network-entry-ack IE on NCFG TxOpps and then cancels the temporary minislot allocation reserved by it for the new node.

Notice that, for a node that just joined the network, it is only allowed to transmit its MSH-NCFG/MSH-DSCH messages in the TD where its chosen sponsoring node resides. The reason is that at this stage it is possible that only the nodes in the two-hop neighborhood of its chosen sponsoring node know the existence of this new operational node. If this new operational node transmits its control messages in other TDs, nodes in other TDs may not know on which TxOpps they should point their antennas toward the TDs where this new operational node is. Worse yet, they may schedule their own control message transmissions on the same TxOpps used by the new operational node, generating inter-node scheduling conflicts.

To solve this problem, we require a new operational node not to transmit its own control messages to TDs where nodes have not learned its existence. The new operational node should continuously transmit MSH-NENT messages (without containing any network-joining messages) to each of its TDs (excluding the TD where its sponsoring node is) within each of the k-NENT-TxOpp groups in the probabilistic manner described above. Only when it finds the MSH-NCFG/MSH-DSCH messages transmitted from the same node in a TD contain its next NCFG/DSCH TxOpp information (indicating that this new operational node is in the unknown scheduling status and considered to contend for every NCFG/DSCH TxOpp) **twice**, can it stop transmitting its MSH-NENT message in this TD and start scheduling its next NCFG/DSCH TxOpp in this TD. This is because now the new node can ensure that nodes in this TD know its existence and have notified their one-hop neighboring nodes of its existence. Thus, the transmissions of its MSH-NCFG/MSH-DSCH messages in this TD will not lead to any inter-node scheduling conflicts. If the new node does not receive any MSH-NCFG/MSH-DSCH messages in a TD after k^2 NENT TxOpps, it means that no operational node is present in this TD. Thus, the new node can start scheduling its MSH-NCFG/MSH-DSCH message transmissions in this TD. Such control message transmissions are still needed because there may be several new nodes in this TD.

3.2.5 Problem 4: Transmission-domain-aware Minislot Scheduling

The minislot scheduling among nodes in the IEEE 802.16(d) mesh CDS mode is negotiated using THP standardized in [1]. However, the scheduling algorithm for allocating minislots in this network is not standardized. Thus, a minislot scheduling algorithm that can exploit the spatial-reuse property of single-switched-beam antennas is desired for 802.16(d) mesh CDS-mode networks using such antennas. In this section, we propose an easy-to-implement Transmission-domain-aware Minislot Scheduling Algorithm (TMSA) which need not modify the messages used in the original standardized THP.

In TMSA, for a node x, there are four types of minislot allocations: the first one is the *"local node transmission"* type (denoted as "LOCAL_XMIT"), indicating that a minislot allocation is used by node x to transmit data packets to its neighboring node; the second one is the *"local node reception"* type (denoted as "LOCAL_RECV"), indicating that a minislot allocation is used by node x to receive data packets from its neighboring node; the third one is the *"neighboring node transmission"* type (denoted as "NBR_XMIT"), which indicates that a minislot allocation is used by a node x's neighboring node to transmit data packets to another neighboring node (other than node x); and the final one is the *"neighboring node reception"* type (denoted as "NBR_RECV"), which indicates that a minislot allocation is used by a node x's neighboring node to receive data from another neighboring node (other than node x). For brevity, in the following we use the notation $(x \rightarrow y)$ to represent a minislot allocation that is used by node x to transmit data packets to node y.

Consider Fig. 3.15, which shows an example case where data packets get collided in a network using omnidirectional antennas. In this example case, node B schedules a minislot allocation (B→A) to transmit its data packets to node A. At the same time, node D also schedules a minislot allocation (D→C) to transmit its data packets to node C. Node C will thus receive data packets from nodes B and D at the same time, resulting in packet collisions on node C. Due to this reason, such a minislot scheduling is not allowed in an omnidirectional-antenna network

In contrast, in a network using single-switched-beam antennas such a minislot scheduling is allowed. Consider a network using single-switched-beam antennas shown in Fig. 3.16. Due to the directivity of switched-beam antennas, the minislot allocations (B→A) and

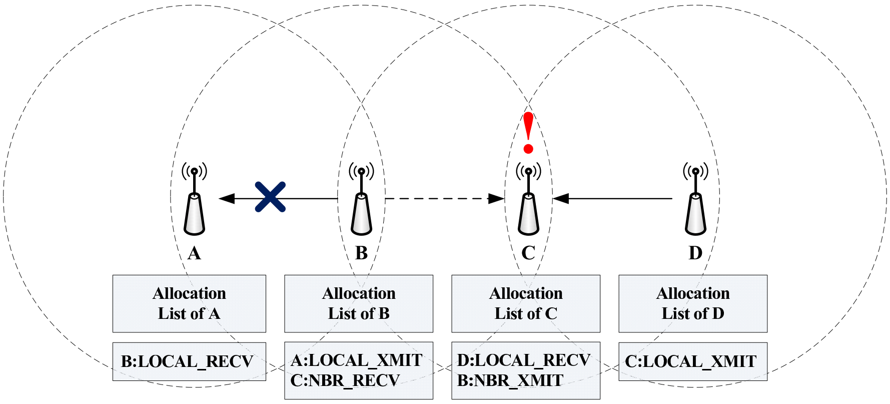


Figure 3.15: A minislot scheduling example in a network using omnidirectional antennas

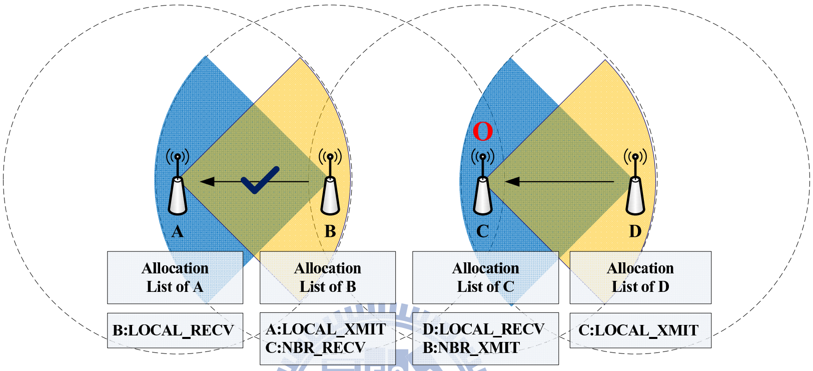
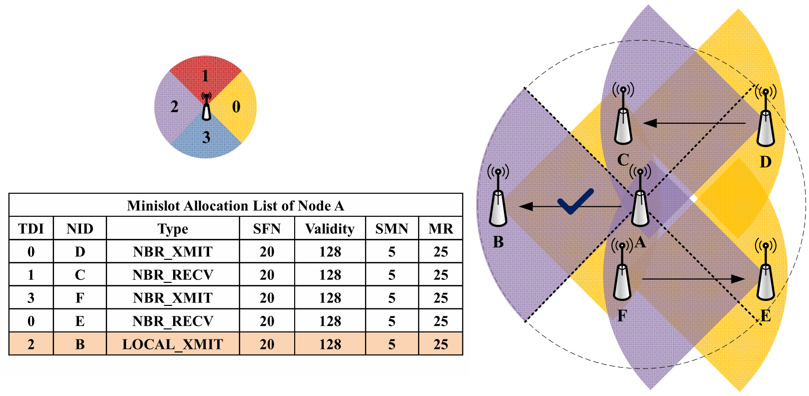


Figure 3.16: A minislot scheduling example in a network using single-switched-beam antennas

(D→C) are allowed to coexist at the same time without generating packet collisions. The reason is that, in the duration when (B→A) and (D→C) simultaneously take place, node B can point its antenna towards the TD covering node A, whereas node D can point its antenna towards the TD covering node C. Under such an arrangement, packets transmitted from these two nodes will not collide with each other on node C, allowing the coexistence of the minislot allocations (B→A) and (D→C).

In TMSA, the description of a minislot allocation is represented by a 7-tuple pair (TDI, NID, type, SFN, validity, SMN, MR), where the TDI field denotes the Transmission Domain Index. The type field denotes the type of this minislot allocation; SFN denotes the starting frame number of this minislot allocation, validity denotes the number of frames that this minislot allocation lasts; SMN denotes the starting minislot number of this minislot allocation within a frame; and MR (Minislot Range) denotes the number of minislots occupied by this minislot allocation within a frame. The interpretation of the NID field depends on the value of the type field. The NID field represents the ID of the



Minislot Allocation List of Node A						
TDI	NID	Type	SFN	Validity	SMN	MR
0	D	NBR_XMIT	20	128	5	25
1	C	NBR_RECV	20	128	5	25
3	F	NBR_XMIT	20	128	5	25
0	E	NBR_RECV	20	128	5	25
2	B	LOCAL_XMIT	20	128	5	25

Figure 3.17: An example of data scheduling in a network using single-switched-beam antennas

receiving node when the following type field is "LOCAL_XMIT." Otherwise, it represents the ID of the transmitting node. The allocations of "NBR_XMIT" and "NBR_RECV" types are learned from received MSH-DSCH messages.

We use the example shown in Fig. 3.17 to illustrate the operation of TMSA. In this example, node D has scheduled a minislot allocation (D→C) that occupies minislots ranging from 5 to 29 during the frames from 20 to 147 and node F has scheduled a minislot allocation (F→E) that occupies the same duration. From the perspective of node A, the former minislot allocation is represented as two 7-tuple pairs (0, D, NBR_XMIT, 20, 128, 5, 20) and (1, C, NBR_RECV, 20, 128, 5, 20), and the latter minislot allocation is represented as another two 7-tuple pairs (3, F, NBR_XMIT, 20, 128, 5, 20) and (0, E, NBR_RECV, 20, 128, 5, 20).

Suppose that node A wants to schedule its minislot allocation (A→B) in the same duration. Using TMSA, node A first checks its minislot allocation list to see whether there is any minislot allocation with the LOCAL_XMIT or LOCAL_RECV type in that duration. If yes, it cannot schedule any minislot allocation in this duration because it has only one antenna and cannot simultaneously transmit or receive packets in different TDs. In this condition, node A should try to schedule (A→B) within a different duration.

In case no such minislot allocation exists in the list, node A then checks whether the list contains any learned NBR_XMIT allocations with node B being the transmitting node for the same duration. If yes, it should not schedule (A→B) because node B cannot receive its packets in that duration. If no, node A finally checks whether the list contains a learned NBR_RECV allocation in the TD where node B resides for the same duration.

If yes, it should not schedule (A→B); otherwise, this transmission will interfere with the neighboring node's data packet reception. If no, (A→B) can be scheduled. In this example, because no such an allocation exists, node A can schedule this (A→B) at the same time when (D→C) and (F→E) take place.

From this example, one sees that the proposed TMSA can utilize the spatial-reuse advantage of a network using single-switched-beam antennas to increase a network's capacity on the data plane. This is achieved with the help of the MTD scheme, which operates on the control plane to ensure that there is no control message collisions in an IEEE 802.16(d) mesh CDS-mode network operating with single-switched-beam antennas.

Chapter 4

Performance Evaluation

4.1 Analysis for the Omnidirectional-antenna Network

In this section, we analyze the Average Three-way Handshake Procedure Time (denoted as ATHPT) in networks using fixed holdoff time schemes and those using our proposed schemes. We derive the ATHPT value based on the used holdoff time value and the number of 1-hop and two-hop neighboring nodes. Because ATHPT greatly affects when a node can transmit data, the analyses presented in this section greatly help us to obtain more insights into the performance of an IEEE 802.16 mesh CDS-mode network.

4.1.1 ATHPT in Networks using Identical Holdoff Time

Fig. 4.1 shows the transmission cycle of a node. Denote H_k as the holdoff time of node k and C_k as the contention time of node k. Denote the interval between two consecutive MSH-DSCH message transmissions for node k (i.e., the interval between two consecutive TxOpps won by node k) as τ_k, which is defined as follows:

$$\tau_k = H_k + C_k. \tag{4.1}$$

Suppose that the scheduling information of nodes are known to each other. That is, all nodes are one-hop neighboring to each other. Further assume that each node uses the same holdoff time (i.e., $H_k = H$, $\forall k \geq 0$, which is the default holdoff time setting in the

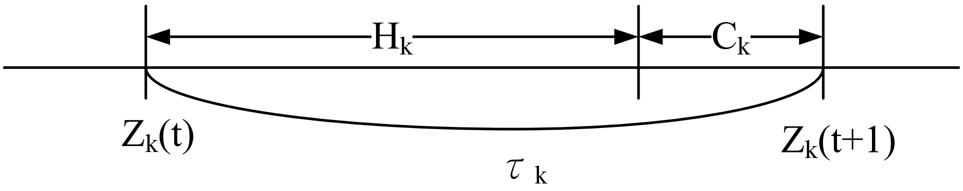


Figure 4.1: The transmission cycle of a node

IEEE 802.16 mesh CDS-mode network). In a network where $nbr_2(k) \leq (2^{base+exp} - 2^{exp})$, each node suspends itself for a sufficiently long interval; thus, it can win a TxOpp just after its holdoff time elapsed. (That is, C_k is only 1.) In this condition, τ_k is:

$$\tau_k = H + 1. \tag{4.2}$$

The proof is given below.

Theorem 4.1.1. *Let $G = (V, E)$ be an IEEE 802.16 mesh CDS-mode network, where each node $v_i \in V$ is 1-hop neighboring to each other and each v_i uses the same holdoff time $H = 2^{exp+base}$. Suppose that $|V| \leq (2^{exp+base} - 2^{exp})$. Then, each node v_i wins a TxOpp every $(H + 1)$ TxOpps.*

Proof. Let n be the number of nodes in the network (i.e., $n = |v|$) and m be the number of MSH-DSCH TxOpps in the interval of $(2^{exp+base} - 2^{exp})$. The MEA of the IEEE 802.16 mesh CDS-mode network should choose the winning node for each TxOpp from the eligible node list of that TxOpp. The chosen eligible node cannot voluntarily yield this TxOpp. Thus, if $n \leq m$ and the network is stationary, the permutation of the winning nodes of m consecutive MSH-DSCH TxOpps is $n!$, followed by $(m - n)$ idle MSH-DSCH TxOpps.

This proves that, in this condition, after node v_i wins a TxOpp α, all other nodes will win their respective TxOpps in the interval $[\alpha + 1, \alpha + H]$. Because all nodes use the same holdoff time, the earliest TxOpp for which nodes other than v_i can contend is $(\alpha + 1) + (H + 1) = \alpha + H + 2$. For this reason, the only eligible node on TxOpp $(\alpha + H + 1)$ (which is the earliest TxOpp for which node v_i will contend again) is node v_i itself. Therefore, node v_i wins a TxOpp every $(H + 1)$ TxOpps. □

Notice that, in the IEEE 802.16 mesh CDS mode, MSH-DSCH TxOpps are contained only in schedule control sub-frames. $4 * N_{sf}$ frames containing a schedule control sub-frame follows a frame containing a network control sub-frame, where N_{sf} denotes the number of frames containing schedule control sub-frames between two frames containing network control sub-frames in multiples of 4. Thus, the interval between two MSH-DSCH message transmissions taking into account the frame usage of the network for node k is given as follows:

$$\phi_k = \tau_k * T_{fr} * \{\frac{\tau_k}{N_{dsch}} + \frac{\tau_k}{4 * N_{sf} * N_{dsch}}\} = \tau_k * \frac{T_{fr}(4N_{sf}+1)}{4N_{sf}N_{dsch}}, \tag{4.3}$$

where N_{dsch} denotes the number of MSH-DSCH TxOpps per frame and T_{fr} denotes the length of a frame in milliseconds.

Combining Eq. 4.1 and Eq. 4.3, we obtain ϕ_k as follows:

$$\phi_k = (H+1) * \frac{T_{fr}(4N_{sf}+1)}{4N_{sf}N_{dsch}}. \tag{4.4}$$

As can be seen, each node can win a TxOpp every $(H+1)$ TxOpp. Thus, in a THP, after the requesting node broadcasts its request IE out, the granting node must be able to broadcast its granting IE out before the next TxOpp won by the requesting node arrives. This means that the requesting node can complete a THP within an interval of ϕ_k. Thus, we can obtain that ATHPT $= \phi_k$, if $nbr_2(k) \leq (2^{base+exp} - 2^{exp})$. The formal proof is given below.

Theorem 4.1.2. *Let $G = (V, E)$ be an IEEE 802.16 mesh CDS-mode network, where each node $v_i \in V$ is 1-hop neighboring to each other and each v_i uses the same holdoff time $H = 2^{exp+base}$. Suppose that $|V| \leq (2^{exp+base} - 2^{exp})$. Then, a Three-way Handshake Procedure (THP) is finished within $(H+1)$ TxOpps.*

Proof. Suppose that node x intends to perform a THP with node y. Theorem 4.1.1 proves that, after node x wins a TxOpp α, all other nodes including node y will win their respective TxOpps in the interval $[\alpha+1, \alpha+H]$. Thus, after node x transmitted its request IE on TxOpp α, node y can transmit its grant IE within the interval $[\alpha+1, \alpha+H]$. Finally, node x can transmit its confirm IE on TxOpp $(\alpha+H+1)$. As a result, a THP can be completed using only $(H+1)$ TxOpps. □

4.1.2 ATHPT in Networks using Non-identical Holdoff Times

Denote the requesting node and the granting node in a THP as N_R and N_G, respectively. The holdoff times of N_R and N_G are denoted as H_R and H_G, respectively, and the contention times of these two nodes are denoted as C_R and C_G, respectively. C_R and C_G are random variables.

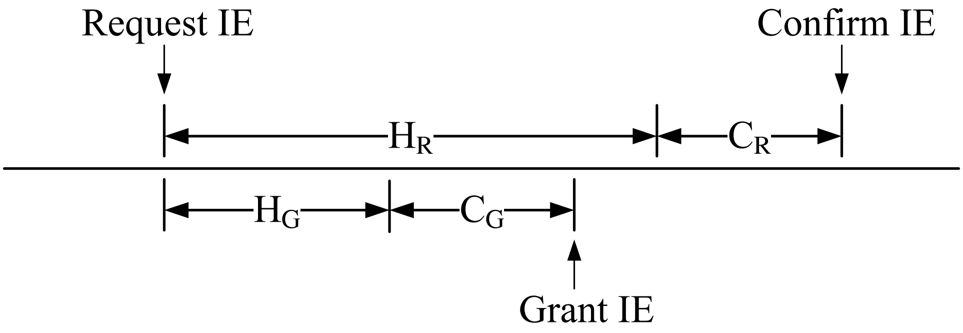


Figure 4.2: Example case where $H_R \gg H_G$

If $H_R \gg H_G$ (as shown in Fig. 4.2), then ATHPT is dominated by the transmission cycle of the requesting node and can be defined as follows:

$$\text{ATHPT} = H_R + E[C_R]. \tag{4.5}$$

Deriving $E[C_R]$ is difficult because it is involved with the holdoff times used by neighboring nodes. In [2], however, *Cao et al.* show that the distribution of a node's contention time can be approximately modeled by a geometric distribution. Following this assumption, $E[C_R] = \frac{1}{p}$, where p is the probability for a node to win a TxOpp. Since, for node x, the maximum number of nodes contending for a TxOpp is $|nbr_2(x)|$, p is given as follows:

$$p(x) \geq \frac{1}{|nbr_2(x)|}. \tag{4.6}$$

We take the lower-bound value $\frac{1}{|nbr_2(x)|}$ for $p(x)$ to estimate the lower-bound performance of a network using static non-identical holdoff times. The value of ATHPT in such a network is thus derived as follows:

$$\text{ATHPT} = H_R + nbr_2(N_R). \tag{4.7}$$

As shown in Fig. 4.3, if $H_R \ll H_G$, then ATHPT $= e_G + e_R$, where e_G denotes the residual time of τ_G and e_R denotes the residual time of τ_R, where $\tau_G = H_G + C_G$ and $\tau_R = H_R + C_R$. Thus, the ATHPT value in this condition can be defined as follows:

$$\text{ATHPT} = E[e_G] + E[e_R]. \tag{4.8}$$

To simplify the analysis, we adopt the assumptions used in [2], which is defined here. Let $Z_k(t)$ be the number of transmission times of node k up to slot t. That is, $Z_k(t)$ is a counting process with inter-arrival time τ_k. Suppose that $\{\tau_k = H_k + C_k, \ \forall k \geq 0\}$ is independent and identically distributed (i.i.d.). Then, $Z_k(t)$ is a renewal process and τ_k

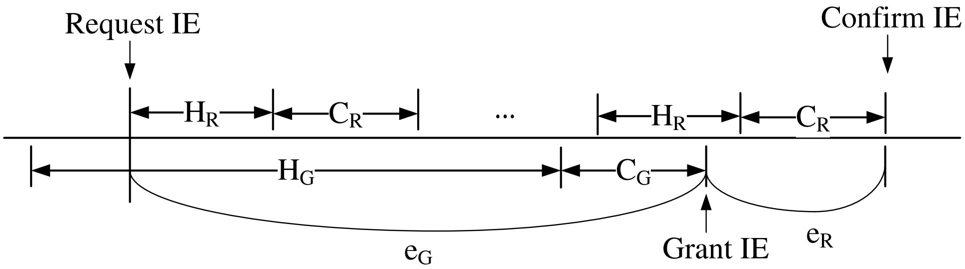


Figure 4.3: Example case where $H_R \ll H_G$

is the renewal interval. With this assumption, we know that the expected value of the residual time of next arrival is one half of the expected value of the observed inter-arrival time (denoted as τ_{k0}) [18]. The formal expression is given as follows:

$$E[e_k] = \frac{\tau_{k0}}{2}, \tag{4.9}$$

where $\tau_{k0} \geq H_k + E[C_k] = H_k + nbr_2(k)$.

Following this, ATHPT can be defined as:

$$\text{ATHPT} \geq \frac{H_G + nbr_2(N_G)}{2} + \frac{H_R + nbr_2(N_R)}{2}. \tag{4.10}$$

If $H_R \simeq H_G$, the value of ATHPT depends on whether $(H_R + C_R)$ is larger than e_G. If this condition is satisfied, a THP can be completed within a transmission cycle of the requesting node. Otherwise, a THP should be completed using the time amount of $(e_G + e_R)$.

$$\text{ATHPT} = \begin{cases} H_R + E[C_R], & \text{if } (H_R + E[C_R]) > e_G \\ E[e_G] + E[e_R] + \epsilon, & \text{if } (H_R + E[C_R]) \leq e_G \end{cases} \tag{4.11}$$

The results derived from this case (Eq. 4.11) perfectly match the results derived from the cases where $H_R \gg H_G$ and $H_R \ll H_G$. Thus, by combining these equations, we can obtain ATHPT as follows:

$$\text{ATHPT} = \begin{cases} H_R + nbr_2(N_R), & \text{if } (H_R + E[C_R]) > e_G \\ \frac{H_G + nbr_2(N_G) + H_R + nbr_2(N_R)}{2} + \epsilon, & \text{if } (H_R + E[C_R]) \leq e_G \end{cases} \tag{4.12}$$

4.1.3 ATHPT in Networks using Dynamic Holdoff Times

Our proposed dynamic holdoff time scheme makes the requesting node win a TxOpp later than and closest to the next TxOpp won by the granting node. Thus, as shown in Fig. 4.4, using this design ATHPT can be modeled as follows:

$$\text{ATHPT} = E[e'_G] + E[e'_R], \tag{4.13}$$

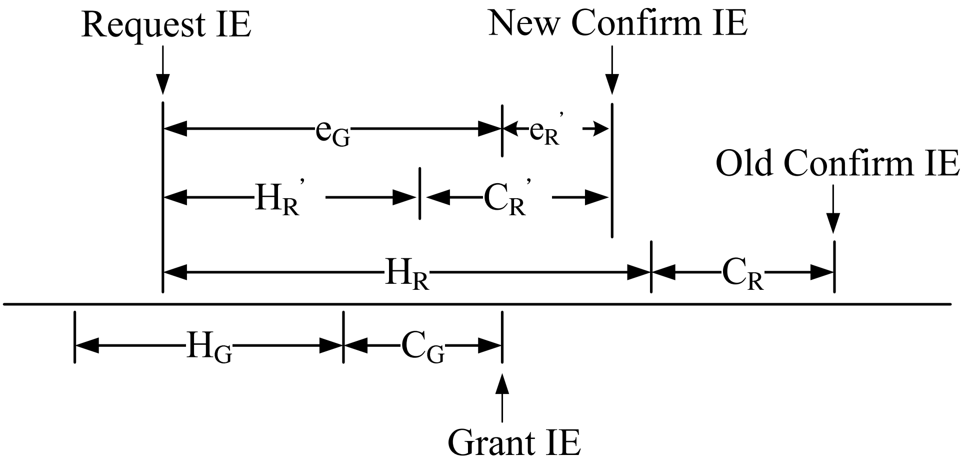


Figure 4.4: Example case where $H_R \gg H_G$ when using the proposed dynamic holdoff time scheme

where $E[e'_G]$ denotes the expected value of the residual time of the granting node's next TxOpp and $E[e'_R]$ denotes the expected value of the residual time of the requesting node's next TxOpp. Similar to $E[e_G]$ in Eq. 4.9, $E[e'_G]$ basically can be modelled as follows:

$$E[e'_G] \geq \frac{H_G + nbr_2(N_G)}{2}. \tag{4.14}$$

However, our proposed dynamic holdoff time scheme makes the holdoff time of a node depend on those of its neighboring nodes. Thus, assuming that all nodes in the network have data to send at all time, we can know that the H_G in Eq. 4.14 is not fixed and approaches to the minimum value of nodes' holdoff times (denoted as H_{min}) in the network, if the network is in the stationary and steady state. Thus, $E[e'_G]$ is corrected as follows:

$$E[e'_G] \geq \frac{H_{min} + nbr_2(N_G)}{2}. \tag{4.15}$$

On the other hand, $E[e'_R]$ can be defined as follows:

$$E[e'_R] = E[H'_R] - E[e_G] + nbr_2(N_R). \tag{4.16}$$

Because in the dynamic holdoff time scheme the requesting node can find a TxOpp later than and closest to the next TxOpp won by the granting node, H'_R is either $2^{\lfloor \log_2(e_G) \rfloor}$ or $2^{\lceil \log_2(e_G) \rceil}$. Thus, $|H'_R - e_G| \leq 2^{\lceil \log_2(e_G) \rceil} - 2^{\lfloor \log_2(e_G) \rfloor}$. Combining these results, ATHPT can be written as follows:

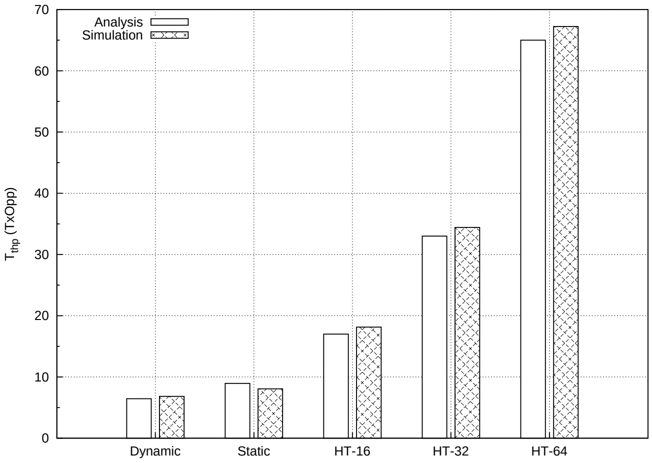


Figure 4.5: Comparison between ATHPT values derived from the theoretical model and those obtained by simulations

$$\begin{aligned}
\text{ATHPT} &\leq \frac{H_{min} + nbr_2(N_G) + nbr_2(N_R) + 2^{\lceil \log_2(e_G) \rceil} - 2^{\lfloor \log_2(e_G) \rfloor}}{2} + \epsilon && (4.17)\\
&\leq \frac{H_{min} + nbr_2(N_G) + nbr_2(N_R) + 2^{\lceil \log_2(\frac{H_{min}+nbr_2(N_G)}{2}) \rceil} - 2^{\lfloor \log_2(\frac{H_{min}+nbr_2(N_G)}{2}) \rfloor}}{2} + \epsilon. && (4.18)
\end{aligned}$$

Because $2^{\lceil \log_2(H_{min}+nbr_2(N_G)) \rceil} - 2^{\lfloor \log_2(H_{min}+nbr_2(N_G)) \rfloor} \leq 2^{\lfloor \log_2(H_{min}+nbr_2(N_G)) \rfloor}$, we can derive ATHPT as follows:

$$\text{ATHPT} \leq \frac{H_{min} + nbr_2(N_G) + nbr_2(N_R) + 2^{\lfloor \log_2(H_{min}+nbr_2(N_G)) \rfloor - 1}}{2} + \epsilon. \qquad (4.19)$$

Eq. 4.19 can be used to evaluate ATHPT. We compare the ATHPT values derived from this analytical model and those obtained from simulations in a 10-node chain network. The results are plotted in Fig. 4.5. As can be seen, the results shown in Fig. 4.5 shows that the ATHPT values derived from our analytical model greatly matches those obtained by the simulation experiments.

4.2 Numeric Evaluation for the Omnidirectional-antenna Network

4.2.1 Performance Metrics

In the following, several performance metrics used throughout the paper are defined.

The Average Transmission Opportunity Utilization of Nodes (ATOUN)

The utilization of a node's control-plane bandwidth is an important metric used to evaluate the efficiency of a holdoff time setting scheme. To define the utilization of the control-plane bandwidth from the perspective of an SS node, one first recalls the notion of the two-hop neighborhood (Eq. 2.1). For an SS node, since the *hidden terminal* problem can occur only with nodes that are in its two-hop neighborhood, the IEEE 802.16 standard requires that each node resolve the contention of each transmission opportunity with the nodes in its two-hop neighborhood. Thus, within a node's two-hop neighborhood, only one node can transmit a control message at any given transmission opportunity.

The average transmission opportunity utilization viewed from a node j is defined in (4.38). This definition indicates how well the nodes in node j's two-hop neighborhood (including node j itself) together utilize the network's transmission opportunities. Ideally, the average transmission opportunity utilization viewed from each node should be 100%, which indicates that, from the perspective of each node, each transmission opportunity is used by one and only one node and no transmission opportunity is left unused.

The ATOUN metric of a network case is defined in (4.40). It is the average across all nodes' AvgTxOpp values in a network case.

$$AvgTxOpp(j) = \frac{\sum_{i \in nbr(j)} txnum(i)}{total(j)} \tag{4.20}$$

$$ATOUN = \frac{\sum_{j=1}^{m} AvgTxOpp(j)}{m} \tag{4.21}$$

where $txnum(j)$ denotes the number of transmission opportunities won by node j, $total(j)$ denotes the number of total transmission opportunities since node j has attached itself to the network, and m is the number of nodes in a network case.

The ATOUN metric reflects the utilization of the control-plane bandwidth from the aggregate of the local view of each node. A higher value of this metric indicates that a network case has a higher control-plane bandwidth utilization; a lower value indicates that a network case has a lower control-plane bandwidth utilization.

The ATOUN metric does not measure the fairness of bandwidth sharing in a network. To solve this problem, we designed another metric, explained in Section 4.2.1, to evaluate how fairly network nodes share the control-plane bandwidth.

The Average Three-way Handshake Procedure Time (ATHPT)

The average three-way handshake procedure time (ATHPT) metric is defined as the average time required by the three-way handshake procedure to establish a data schedule across all network nodes in a case. This metric is computed as follows. For a network case, we first use (4.41) to average the times required to establish data schedules for every node. We then use (4.42) to compute the case's ATHPT value, which is the average across all nodes' THPT values. Like the ATOUN metric, for each scheme, the average and standard deviation of its ATHPT values across all simulation cases will be presented.

$$THPT(j) = \frac{\sum_{i=1}^{n} t_{ij}}{n} \tag{4.22}$$

$$ATHPT = \frac{\sum_{j=1}^{m} THPT(j)}{m} \tag{4.23}$$

where t_{ij} denotes the time required for establishing the i_{th} data schedule of node j, n is the number of node j's data schedules, and m is the number of nodes in a network case.

ATHPT is a common metric used in the literature to evaluate the effect of the holdoff time value. The three-way handshake procedure requires transmitting three MSH-DSCH messages, each of which contains the request, grant, and confirm information elements (IE), respectively. The detailed procedure is described below.

First, the requesting node transmits a request IE to the peer node. The request IE specifies (1) the number of requested mini-slots on the peer node and (2) the available mini-slots on the requesting node from which the peer node can choose. On receiving the request IE, the peer node decides whether it would like to accept this request. If not, it ignores this message. Otherwise, out of its own available mini-slots, it allocates a data schedule from the requesting node's available mini-slots. The peer node then transmits a grant IE containing the information of the allocated data schedule to the requesting node. Upon receiving the grant IE, the requesting node broadcasts a confirm IE to all of its neighboring nodes to notify them of this allocation information.

The reasons for the above procedure are clear. First, the mini-slots of the requesting and the peer nodes are already synchronized over the time axis in an IEEE 802.16 mesh network. Second, when the requesting node is transmitting data to the peer node, the peer node must be able to receive the data at the same time. Therefore, the requesting node must negotiate with the peer node to find a range of mini-slots that is available to

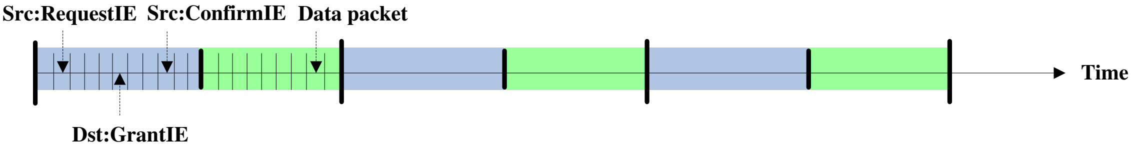


Figure 4.6: A good case for establishing a data schedule in the distributed coordinated scheduling mode when the network is not congested

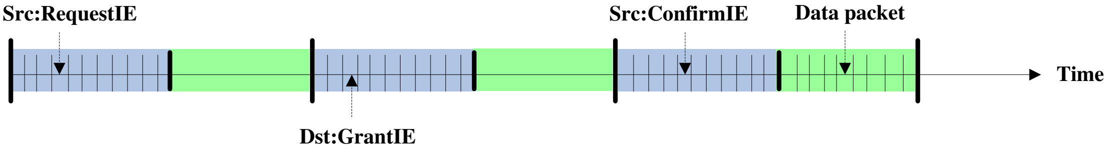


Figure 4.7: A bad case for establishing a data schedule in the distributed coordinated scheduling mode when the network is not congested

both of them (i.e., good for transmitting at the requesting node and good for receiving at the peer node) and can accommodate the requested number of mini-slots.

Fig. 4.6 and Fig. 4.7 are two examples showing the effect of the holdoff time value. (Note that these two figures are for illustration purposes and the minimum number of slots between subsequent MSH-DSCH messages should be 16 according to the standard.) Fig. 4.6 shows a good case for establishing a data schedule when the network is not congested. In this case, all control messages required for establishing a data schedule are exchanged within one control subframe due to the use of a small holdoff time value. Thus, data packets can be quickly transmitted within the same frame in which the control messages are transmitted. In contrast, Fig. 4.7 shows a bad case for establishing a data schedule when the network is not congested. In this case, the three control messages are transmitted over three different frames due to the use of a large holdoff time value. In such a condition, the data packets can only be transmitted over the mini-slots that are at least two frames away from the transmission of the Request IE. Since a node is allowed to transmit data packets to its neighboring node only after they have established a data schedule, the increased delay of the three-way handshake procedure directly degrades the network quality experienced by application programs.

Network Unfairness Index (NetUI) and Node Unfairness Index (NodeUI)

We define two new performance metrics, named *"Network Unfairness Index (NetUI),"* and *"Node Unfairness Index (NodeUI),"* to evaluate how fairly network nodes share the control-plane bandwidth. We explain the definitions of these two metrics here. To understand them, one first realizes that, viewed from a node, if every node in its two-hop neighborhood has data to send at any given time, the optimal way to schedule these nodes' control message transmissions is to schedule them in a round-robin fashion. That is, a node should on average transmit one and only one control message every N TxOpps, where N is the number of nodes in its two-hop neighborhood (including itself). We call this round-robin scheme "the static optimal scheme" in this dissertation. This is the optimal design for a static network in which every node has data to send at all time. This is because when a node wants to transmit a control message, the transmission must be resolved among all the nodes in its two-hop neighborhood. Thus, to avoid congestion while reducing transmission delays, on average a node can only transmit a control message every N TxOpps, where N is defined above.

For a fixed-value holdoff time scheme, to avoid any congestion from occurring in the network, the maximum of the N values of all nodes should be used as the fixed value for all nodes. Since in a general network topology not all nodes have the same N value, this fixed-value approach will waste the transmission opportunities of the nodes whose N values are smaller than the maximum one.

We define a node's TxOpp utilization during a period as the ratio of the number of TxOpps that it wins during the period to the total number of TxOpps available during that period. If such ratios of all nodes under a holdoff time scheme closely approximate their counterparts under the static optimal scheme, this holdoff time scheme is considered to perform as well as the static optimal scheme.

The following explains the steps used to compute NetUI. First, we convert the actual utilization ratio of a node i into the logarithmic form as follows:

$$R1(i) = -log_2(\frac{NumTxopp_{win}(i)}{NumTxopp_{total}}) \tag{4.24}$$

where $NumTxopp_{win}(i)$ denotes the number of TxOpps that node i wins in the period and $NumTxopp_{total}$ denotes the total number of TxOpps available in the period. Also, the logarithmic form of the static optimal utilization ratio for a node i is shown in the following:

$$R2(i) = -log_2(\frac{1}{|nbr(i)|}) \tag{4.25}$$

where $|nbr(i)|$ denotes the number of nodes in node i' two-hop neighborhood. Second, the absolute value of the difference between the two logarithms is computed and shown as follows.

$$Diff(i) = AbsoluteValue(R1(i) - R2(i)) \tag{4.26}$$

Finally, the NetUI metric is defined as the sum of the Diff values of all nodes in a network and shown as follows:

$$NetUI = \sum_{i=1}^{m} Diff(i) \tag{4.27}$$

where m is the number of nodes in a network case.

The rationale for the NetUI metric is that when evaluating a holdoff time scheme, one should consider its impacts on all network nodes. This sum shows the degree of inefficient and unfair use of available transmission opportunities across all nodes in a network. A zero NetUI value means that a holdoff time setting scheme schedules transmission opportunities for the whole network as if the static optimal scheme were used. In contrast, a non-zero NetUI value indicates that the scheme schedules transmission opportunities either inefficiently or unfairly when compared with the static optimal scheme. As expected, a high NetUI value indicates that the used holdoff time setting scheme deviates much from the static optimal scheme.

The definition of NodeUI is given as follows:

$$NodeUI = \sum_{i=1}^{m} \frac{Diff(i)}{m} \tag{4.28}$$

where m is the number of nodes in a network case.

The NodeUI metric is similar to the NetUI metric. However, it is more suitable for observing the inefficiency and unfairness degree of a node's control message scheduling. The best value for this difference is zero, which means that the scheduling generated by the used scheme for this node is equivalent to that generated by the static optimal scheme. If this difference value increases, it means that the scheme performs worse than the static optimal scheme. In the following sections, either NetUI or NodeUI is used to evaluate the fairness degree of TxOpp sharing in the network, depending on the context.

Table 4.1: The parameter setting used in simulations

Parameter Name	**Value**
MSH-CTRL-LEN	8
MSH-DSCH-NUM	8
Scheduling Frames	2
Requested Mini-slot Size	20
Requested Frame Length	32
Modulation/Coding Scheme	64QAM-3/4
Maximum Transmission Range	500 meter
Frame Duration	10 ms

4.2.2 Simulation Setting

In this section, we evaluate the performances of the static and dynamic approaches. We compare the simulation results of these approaches with those of the fixed-value holdoff time setting schemes with holdoff times being 16, 32, and 64. (denoted as HT-16, HT-32, and HT-64, respectively).

The NCTUns network simulator and emulator [19] is used for the evaluation. The chain, grid, and random network topologies are used for these performance studies. Each reported performance is the average of five runs using different random number seeds. For each run, the simulated time is 1,000 seconds. Table 4.1 shows the parameter setting used in our simulations. More detailed setup specific to a particular network topology will be described in the following subsections.

4.2.3 Chain Network Topology

The chain network topology is composed of 19 nodes. From left to right, the nodes are named BS, SS(1), SS(2), ..., and SS(18), respectively. On this chain network, each node runs a MAC-layer pseudo data scheduler to periodically establish data schedules with its neighboring nodes in a round-robin manner. The frequency is chosen to be one data schedule every 3 seconds. The duration of a minislot allocation is set to 32 frames.

As shown in Table 4.2(a), in the chain network, as the holdoff time value exponentially increases, the ATOUN value exponentially decreases and the ATHPT value exponentially increases. These results show that when the holdoff time value exponentially increases, the average transmission opportunity utilization significantly decreases and the average three-way handshake procedure time significantly increases. The reasons for these phenomena

Table 4.2: The performances of the evaluated schemes

(a) Chain Network Topology

	MAC						Application					
	ATOUN		ATHPT (ms)		NetUI		TCP (KB/sec)		UDP (KB/sec)		Ping (ms)	
	Avg.	Std.dev.	Avg.	Std.dev.	Avg	Std.dev.	Avg.	Std.dev.	Avg.	Std.dev.	Avg	Std.dev.
Dynamic	0.641	0.1140	14.295	12.2361	8.292	0.3353	381.812	160.7230	573.201	3.6810	390.030	224.1563
Static	0.494	0.0830	17.389	14.9974	18.140	3.553e-15	372.268	161.4577	569.606	4.4822	410.817	234.4550
HT-16	0.210	0.0330	24.184	5.4280	46.271	2.132e-14	235.043	171.8394	534.433	5.8250	825.176	466.1554
HT-32	0.110	0.0180	46.285	10.8420	66.365	1.421e-14	149.935	142.9734	472.869	8.5998	1430.85	810.8086
HT-64	0.057	0.0090	89.256	19.4830	86.929	1.499e-3	85.129	102.9986	401.746	13.1154	2609.386	1481.6938

(b) Grid Network Topology

	MAC					
	ATOUN		ATHPT (ms)		NetUI	
	Avg.	Std. dev.	Avg.	Std. dev.	Avg.	Std. dev.
Dynamic	0.725	0.0860	43.398	9.4300	9.661	0.1460
Static	0.559	0.0822	50.467	11.1090	59.303	0.0000
HT-16	0.530	0.0900	50.983	8.2110	70.313	0.0000
HT-32	0.405	0.0830	62.852	5.8381	116.974	0.0112
HT-64	0.245	0.0590	104.479	3.6402	199.009	0.0031

(c) Random Network Topology

	MAC					
	ATOUN		ATHPT (ms)		NetUI	
	Avg.	Std. dev.	Avg.	Std. dev.	Avg.	Std. dev.
Dynamic	0.718	0.1290	47.425	25.8386	14.896	0.5942
Static	0.543	0.1121	55.312	31.8472	66.718	1.9031
HT-16	0.481	0.1641	55.194	23.6402	95.336	6.0831
HT-32	0.363	0.1654	66.453	15.7061	150.262	8.2523
HT-64	0.237	0.1313	104.146	7.5184	226.507	10.0772

have been explained before. As for the static and dynamic approaches, one sees that they significantly outperform the three fixed-value holdoff time setting schemes on the ATOUN and ATHPT metrics. One also sees that the dynamic approach outperforms the static approach. This is because the former can dynamically adjust the holdoff time value to reduce the time interval between sending a request IE and sending a confirm IE. As such, the time required for completing a three-way handshake procedure (and thus for establishing a data schedule) can be greatly reduced. This also explains why the dynamic approach generates a higher utilization of transmission opportunity than the static approach.

For NetUI, when the holdoff time value decreases, the NetUI value decreases as well. This result shows that using a smaller holdoff time value can achieve fairer and more efficient scheduling. One sees that the static and dynamic approaches achieve much smaller NetUI values than the fixed-value holdoff time setting schemes. The result is expected as in both approaches, the holdoff time of each node can be independently set to a different value to reflect the node density around it. In contrast, as discussed before, a fixed-value holdoff time setting scheme cannot suit the scheduling needs of all nodes in a network. The dynamic approach outperforms the static approach on NetUI and the reason is explained below. To reduce the time required for the three-way handshake procedure, the dynamic approach uses an iterative algorithm to decrease a node's holdoff time value. As such, the dynamic approach eliminates a part of contention time that the static approach cannot eliminate. This makes the dynamic approach perform more closely to the static optimal scheme than the static approach.

Regarding application performances, on this chain network, we conduct a different set of simulations using three different application-layer traffic: TCP, UDP, and ping. For a studied holdoff time setting scheme, its performances are evaluated on 16 simulation runs, each time using a different random number seed. In each run, a traffic flow (either TCP, UDP, or ping) is set up. The source node of the traffic flow is fixed at SS(2) node while the destination node of the traffic flow is chosen to be SS(i+2) in the i_{th} run.

Fig. 4.8 shows the relationship between the TCP throughput and the hop count, Fig. 4.9 shows the relationship between the UDP throughput and the hop count, and Fig. 4.10 shows the relationship between the end-to-end round trip time measured by the ping program and the hop count, respectively. As shown in Fig. 4.8 and Fig. 4.9, the static and dynamic approaches achieve much higher throughputs than the fixed-value holdoff time setting schemes over all studied hop counts. The RTT results show that the static and dynamic approaches reduce the end-to-end round-trip packet delay significantly, when compared with the three fixed-value holdoff time setting schemes. The results also show that the dynamic approach generates a smaller round-trip packet delay than the static approach. This is expected as the dynamic approach can further reduce the time required for establishing a data schedule than the static approach.

Table 4.2(a) shows the TCP and UDP throughputs and round trip times averaged across all different hop counts for each scheme. According to the average TCP and UDP throughput results, the static and dynamic approaches on average achieve higher TCP and UDP throughputs than the fixed-value schemes. For example, the dynamic approach outperforms the "HT-16" scheme by a factor of 1.624 on TCP throughput and by a factor of 1.073 on UDP throughput, respectively. Regarding the round trip time, the dynamic approach on average reduces the round trip time of "ping" packets by a factor of 2.116 when compared to the "HT-16" scheme.

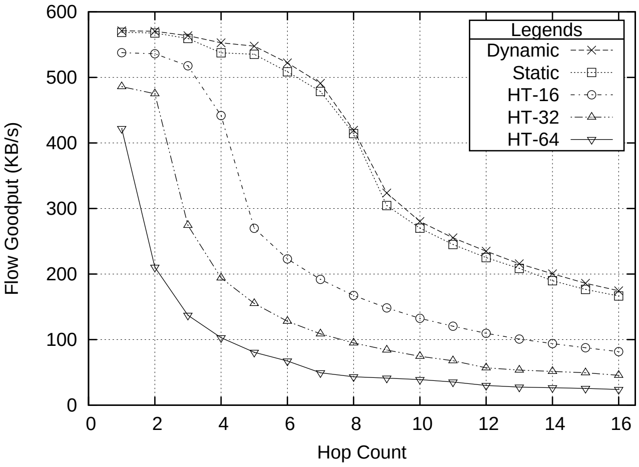


Figure 4.8: TCP throughputs over different hop counts in chain networks

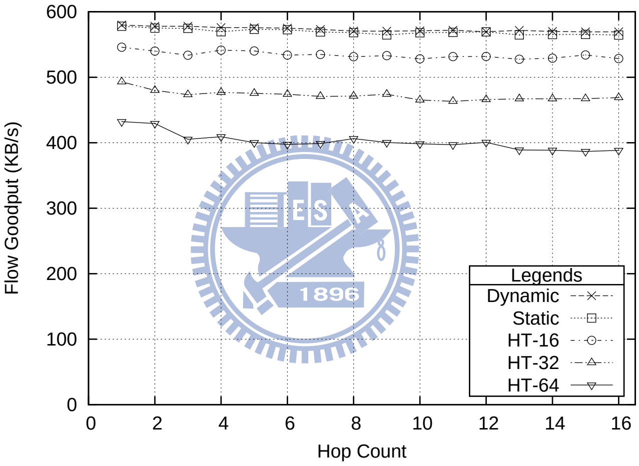


Figure 4.9: UDP throughputs over different hop counts in chain networks

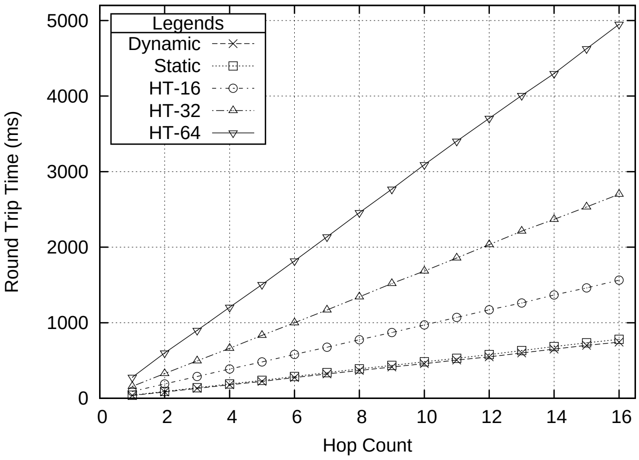


Figure 4.10: The round trip time measured by the ping program in chain networks

4.2.4 Grid Network Topology

For grid network simulations, we construct a 10x10 grid network comprising 100 nodes, each of which is spaced 450 meters apart from its vertical and horizontal neighbors. Each node runs a MAC-layer pseudo data scheduler to periodically establish data schedules with its neighboring nodes in a round-robin manner. As explained previously, the frequency is chosen to be one data schedule every 3 seconds.

As shown in Table 4.2(b), the ATOUN result shows that the dynamic approach achieves the highest utilization of transmission opportunity. The ATHPT result shows that the dynamic approach on average generates the shortest time required for establishing data schedules among all studied schemes. Regarding NetUI, the dynamic approach on average achieves 9.661, which is smaller than the "holdoff time 16" scheme by a factor of 7.278. The NetUI results show that the dynamic approach can both efficiently and fairly schedule transmission opportunities in the control plane. As for the static approach, it on average achieves a better NetUI value than the "holdoff time 16" scheme. However, its performances on ATOUN and ATHPT are close to those of the "holdoff time 16" scheme. The reason is that in a grid network, a node's two-hop neighborhood node number is on average more than 16 and almost every node (except the nodes on the edges of the grid) has the same number. This condition allows the "holdoff time 16" scheme to perform almost equally well with the static approach.

4.2.5 Random Network Topology

In this section, we use ten random connected topologies and derive average simulation results from them to eliminate the effects that may be caused by using a single specific topology. To generate such topologies, we randomly distribute one BS node and 99 SS nodes within a square area with side length of 2,500 meters. We then check whether the generated topology is partitioned or not. If it is partitioned, it is discarded and the above process is repeated until a random connected topology is generated. The whole process is repeated ten times to generate ten random connected topologies. These topologies represent different random dense wireless backbone networks. For each studied holdoff time setting scheme, we conducted its simulations on each topology five times, each time using a different random number seed. Therefore, we have 50 runs in total to derive average simulation results. The simulated time for each run is set to 1,000 seconds. During simulation, each node runs a MAC-layer pseudo data scheduler to periodically establish data schedules with its neighboring nodes in a round-robin manner. The frequency is chosen to be one data schedule every 3 seconds. It generates a moderate traffic load that allows the performances of the studied holdoff time setting schemes to be distinguished.

As shown in Table 4.2(c), in random topologies, both the static and dynamic approaches generate better performances than the fixed-value holdoff time setting schemes. For the static approach, it on average increases the ATOUN value by a factor of 1.128 and decreases the NetUI value by a factor of 1.428, when compared with the "holdoff time 16" scheme. As for the dynamic approach, it on average increases the ATOUN value by 49.27% and decreases the NetUI value by 640.01%, as compared with the "holdoff time 16" scheme. These results show that both approaches generate fairer and more efficient MAC-layer scheduling for the network than fixed-value holdoff time setting schemes.

One sees that the static approach's ATHPT value is close to that of the "holdoff time 16" scheme. This situation is similar to that in the grid network case. In these random topologies, the number of nodes in each node's two-hop neighborhood is on average more than 16. As such, in such a condition, setting all nodes' holdoff time values to 16 is sufficient to decrease the time required for the three-way handshake procedure. As in the chain and grid networks, one also sees that the dynamic approach further reduces the ATHPT value than the static approach.

In summary, these simulation results show that, under most network topologies, the dynamic approach (1) utilizes the control-plane bandwidth efficiently, (2) reduces the time required for establishing data schedules significantly, and (3) utilizes the control-plane bandwidth fairly.

4.2.6 Performance Comparison between the minimum holdoff time Scheme and Our Proposed Scheme

One may think that using the minimum holdoff time (i.e., 1) is an alternative to achieve the optimal network performance. In this section, we compare the performance of such a naive scheme (denoted as the HT-1 scheme) and our proposed dynamic holdoff time scheme. Most of the simulation settings are the same as those used in previous sections, except that the simulated topology is changed to a 25-node grid network, where each is spaced with 450 meters. In the simulated topology, each node establishes a TCP

Table 4.3: MAC-layer performance of the evaluated schemes

	MAC					
	ATOUN		ATHPT (ms)		NodeUI	
	Avg.	Std. dev.	Avg.	Std. dev.	Avg.	Std. dev.
HT-1	0.742	0.0911	19.187	12.9448	0.051	0.0413
Dynamic	0.676	0.0932	18.917	12.9952	0.184	0.0900
HT-16	0.375	0.0787	29.865	11.9366	1.293	0.3121
HT-32	0.223	0.0529	51.028	13.9581	2.131	0.3657
HT-64	0.119	0.0300	97.320	19.2756	3.076	0.3827

connection to each of its one-hop neighboring nodes. The number of frames of each minislot allocation in this series of simulations is set to 128.

Tab. 4.3 shows the MAC-layer performance of the evaluated schemes. As can be seen, both the HT-1 and our proposed dynamic holdoff time scheme can generate the shortest time for completing THPs. However, as compared with the HT-1 scheme, the proposed dynamic holdoff time scheme can further reduce the used DSCH TxOpps. Thus, it is more bandwidth efficient than the HT-1 scheme.

Although both the HT-1 scheme and our proposed dynamic holdoff time scheme can achieve the best network performance among the evaluated schemes, they greatly differ in the required computation complexity. Tab. 4.11 shows 1) the contention time of a node, 2) the size of the contending node list for each TxOpp, and 3) the required computation complexity under the evaluated schemes. The contention time of a node and the average size of the contending node list under the fixed holdoff time schemes are derived based on the analytical model proposed by Cao et al. [2]. The required computation complexity can be estimated by multiplying these two values. Note that, to fairly analyze the contention time and the size of the contending node list, we assume that the next TxOpp information broadcast by each node is the accurate TxOpp number. This can be easily accomplished by adding an offset field into the MSH-NCFG and MSH-DSCH message formats.

As shown in Fig. 4.11, the behaviors of the proposed dynamic holdoff time scheme is more complicated than fixed holdoff time schemes and can be divided into two cases. In the first case, our proposed dynamic holdoff time scheme calculates the most efficient next DSCH TxOpp when a node needs to send a Request IE. This calculation process may need to perform MEA at most three times and the maximum number of contending nodes per TxOpp during this calculation process is $nbr_2(x)$. In the second case, when a node need not transmit Request IEs, our proposed dynamic holdoff time scheme simply

Table 4.4: The computation cost of the HT-1 scheme and the proposed Dynamic holdoff time scheme

	Contention Time	**Contending Node List Size**	**Computation Complexity**
HT-1	$nbr_2(x)$	$nbr_2(x)$	$O((nbr_2(x))^2)$
HT-16	$\frac{2*nbr_2(x)+16}{17}$	$\frac{2*nbr_2(x)+16}{17}$	$O(\frac{4*(nbr_2(x))^2}{289})$
HT-32	$\frac{2*nbr_2(x)+32}{33}$	$\frac{2*nbr_2(x)+32}{33}$	$O(\frac{4*(nbr_2(x))^2}{1089})$
HT-64	$\frac{2*nbr_2(x)+64}{65}$	$\frac{2*nbr_2(x)+64}{65}$	$O(\frac{4*(nbr_2(x))^2}{4225})$
Dynamic (128 frames)	$0.509 * nbr_2(x)$	$0.509 * nbr_2(x)$	$O(0.302 * (nbr_2(x))^2)$
Dynamic (32 frames)	$0.536 * nbr_2(x)$	$0.536 * nbr_2(x)$	$O(0.446 * (nbr_2(x))^2)$
Dynamic (8 frames)	$0.617 * nbr_2(x)$	$0.617 * nbr_2(x)$	$O(0.895 * (nbr_2(x))^2)$
Dynamic (4 frames)	$0.690 * nbr_2(x)$	$0.690 * nbr_2(x)$	$O(1.295 * (nbr_2(x))^2)$
Dynamic (2 frames)	$0.775 * nbr_2(x)$	$0.775 * nbr_2(x)$	$O(1.764 * (nbr_2(x))^2)$
Dynamic (1 frames)	$0.855 * nbr_2(x)$	$0.855 * nbr_2(x)$	$O(2.203 * (nbr_2(x))^2)$

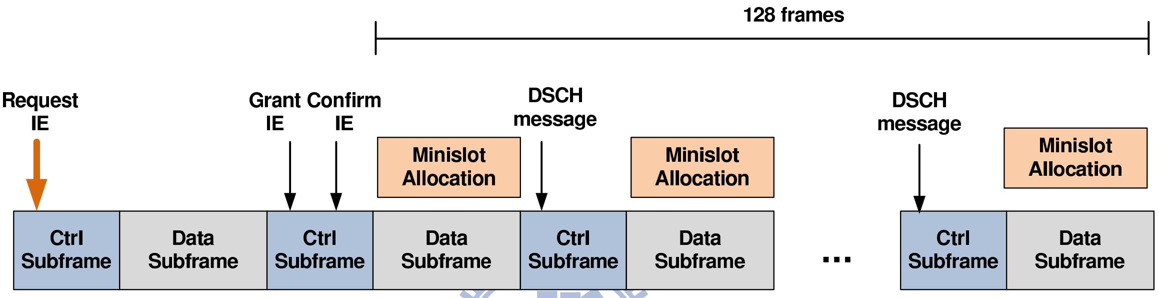


Figure 4.11: The operation of the proposed dynamic holdoff time scheme

sets the node's holdoff time to $2^{floor(log_2(|nbr(i)|))}$. In this condition, the calculation process performs MEA once. We used simulations to estimate the number of contending nodes per TxOpp for all nodes and obtained such a value is roughly $\frac{nbr_2(x)}{2}$.

On the other hand, in this network nodes usually use THPs to reserve a minislot allocation that lasts multiple frames (e.g., 8 frames to 128 frames). This means that the second case more frequently occurs than the first case. For example, suppose that the frame duration of each minislot allocation is 128 frames. In the second case, each node on average transmits its MSH-DSCH messages every 19.6 TxOpps in the 25-node grid network. Given that each scheduling control subframe contains 8 TxOpps, the number of the occurrences of the second cases within 128 frames is 53.895. This means that one occurrence of the first case follows 53.895 occurrences of the second case. Thus, the total computation complexity of the proposed dynamic holdoff time scheme can be amortized by the light-weight processing of the second case. We formalize the expected contention time for node k ($E[C_k]$) as follows:

$$E[C_k] = \frac{1 * nbr_2(x) + l * u}{1 + u}, \tag{4.29}$$

where l denotes the average contention time of node k in the second case (which is $\frac{nbr_2(k)}{2}$ in the 25-node grid network) and u denotes the number of the occurrences of the second case (which is 53.895 when the frame duration is set to 128 frames). Based on Cao's model [2], the expected value of the contending node number per TxOpp is $E[C_k]$. By multiplying these two values, we can obtain the computation complexity of the proposed dynamic holdoff time scheme under different frame durations, which are presented in Tab. 4.4. One sees that, as long as the reserved frame duration is large enough (e.g., larger than 8 frames), the computation complexity required by our proposed scheme is less than that required by the HT-1 scheme.

When the reserved frame duration is very small (e.g., 1 frame to 4 frames), the proposed dynamic holdoff time scheme has higher computation complexity than the HT-1 scheme. This fact inspired us that the proposed dynamic holdoff time scheme can simply set the holdoff time to 1, when the reserved frame duration is small. This is because in the conditions where Request IEs are very frequently transmitted and minislot allocations are very frequently reserved, using large computation cost to reduce the time for scheduling a minislot allocation is not cost-effective.

4.2.7 Summary

In an IEEE 802.16 mesh network, the holdoff time value setting design is very important for achieving good scheduling performances in the distributed coordinated scheduling mode. In this paper, we show that using a small value for this parameter can improve MAC-layer performances, quantified by three performance metrics — ATOUN, ATHPT, and NetUI. However, we show that doing so can easily cause a node's network initialization process to fail in a dense network. In this paper, we further explain why using a fixed holdoff time value for all nodes regardless of their node densities and dynamic bandwidth needs can result in suboptimal performances.

To address these problems, we propose a two-phase holdoff time setting scheme to (1) guarantee the success of network initialization and (2) improve MAC-layer scheduling performances. Both a static approach and a dynamic approach of this scheme are proposed and their performances are studied and compared in this paper.

The overall simulation results show that the dynamic approach significantly outperforms all fixed-value schemes. The ATOUN and ATHPT results show that it significantly increases the utilization of the control-plane bandwidth and decreases the time required for completing the three-way handshake procedure. The NetUI results show that it generates efficient and fair scheduling in the distributed coordinated scheduling mode. In addition, the throughput results show that it generates the highest TCP and UDP throughputs among all studied schemes. Finally, the round-trip time results show that the dynamic approach generates the shortest end-to-end round-trip packet delay among all studied schemes.

4.3 Numeric Evaluation for the Directional-antenna Network

In this section, we use the NCTUns network simulator [19] to evaluate the performances of our proposed scheme. NCTUns is an advanced network simulator with two unique advantages: 1) directly using the real-world TCP/IP protocol stacks in the Linux kernel; and 2) allowing real-world network application programs to run over its simulated networks. As a result, the simulation results generated by NCTUns are generated by real-world traffic generator programs using the real-world Linux TCP/IP protocol stacks. In the following, we describe the simulation settings in Section 4.3.1 and explain the studied performance metrics in Section 4.3.2. Finally, the simulation results are presented in Section 4.3.3.

4.3.1 Simulation Environment

We created a 25-node grid network topology shown in Fig. 4.12 as our simulation topology. The main parameter settings used in the simulations are listed in Tab. 4.5. Two different traffic types are used to generate network traffic. One is greedy TCP and the other is greedy UDP. The word "greedy" means that the source node of a flow will transmit data as many as it can. For example, for a greedy UDP traffic generator program using the standard socket APIs to transmit data, each time when being waken up by the operating system, it will continuously generate data and send them out via socket APIs, until the socket APIs return error codes such as "full socket buffer."

In a simulation case, each node sets up a greedy UDP flow to each of its 1-hop neighboring nodes. Because UDP is a transportation layer protocol that simply transmits data down to the MAC-layer, the throughputs obtained by all greedy UDP flows of a node are equivalent to the MAC-layer throughputs that can be obtained by the nodes minus the bandwidth overheads introduced by MAC-layer, network-layer, and transport-layer headers. Therefore, the throughput results obtained from our greedy UDP flow cases are useful to evaluate the capacity of a network. We use this property to evaluate the capacity gain of an 802.16(d) mesh CDS-mode network using single-switched-beam antennas.

In addition to evaluate the raw network capacity, we also use greedy TCP flows to conduct simulations in the same scenario. TCP is a complicated transportation-layer protocols that employs sophisticated error-control and congestion-control mechanisms to guarantee error-free in-order data delivery. It is widely used by many network applications, such as File Transfer Protocol (FTP) and HTTP web accesses. Thus, observing how TCP works over an 802.16(d) mesh CDS-mode network using single-switched-beam antennas is worthwhile and important to researches for the network transport layer and application development.

Each of our simulation case was run ten times, each time using a different random number seed. The simulated time of each run was set to 500 seconds. In a simulation, the traffic generator programs are activated at the 200-th second of the simulated time. This arrangement is to ensure that they transmit data packets after the simulated network has been stabilized[1]. In each of the figures shown below, both the average and the standard deviation of the collected simulation results are presented.

In the simulated grid network, each node is spaced 450 meters away from its vertical and horizontal neighboring nodes and equipped with a single-switched-beam antenna. The gain pattern of this antenna on the horizontal plane across 360 degrees is plotted in Fig. 4.13, which is derived from [20].

For wireless channel modeling, we conducted simulations with several channel models, such as the two-ray model, Ercegs model with terrain type B, and the ECC-33 model. In our experiences, in absence of dynamic fading effects (e.g., the Rayleigh fading), by properly setting the transmit power of a node, the receive power sensitivity threshold of a

[1] We define a stabilized network as a network in which all of its nodes have joined the network.

Table 4.5: The parameter settings used in the simulations

Parameter Name	**Value**
Number of TxOpps per Frame	8
Number of TxOpps per Frame Used by the CDS Mode	8
Number of Mini-slots per Frame	220
Number of OFDM Symbols per Mini-slot	3
Requested Mini-slot Size per THP	10, 20, 30, 40, 50, 60, 70
Requested Frame Length	2^0, 2^1, 2^2, 2^3, 2^5, 2^7
Modulation/Coding Scheme	64QAM-3/4
Channel Model	ECC-33
Frame Duration	10 ms
TCP Version	Binary Increase Congestion Control TCP (BIC-TCP)

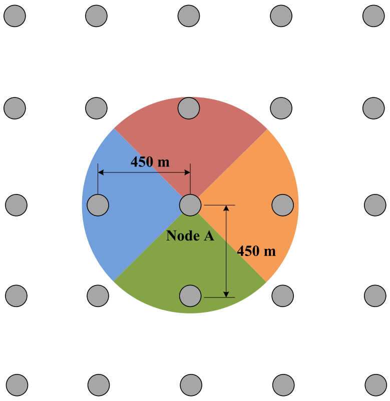


Figure 4.12: The topology of the simulated network

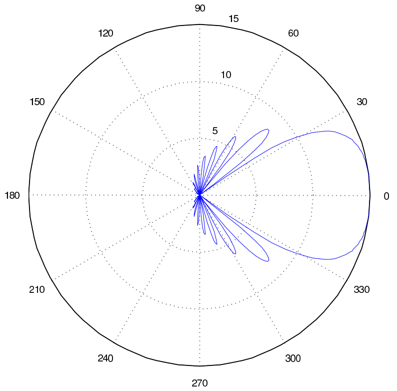


Figure 4.13: The gain pattern of the used single-switched-beam antenna

node, and the distance between nodes, the link connectivity and signal quality experienced by nodes can be adjusted to the same level. Thus, to save space we only present the simulation results under the ECC-33 model in this section. The effects of the Rayleigh fading on network capacity and flow throughputs are studied in Section 4.3.3.

When the dynamic fading effect is absent in our simulations, with proper settings of transmit power and receive sensitivity, the effective transmission range and interference range used in our simulated grid network topology are set to 500 and 850 meters, respectively. In contrast, when the dynamic fading effect is present, there are no explicit values for the transmission range and the interference range of a node. In this condition, a proper receive power threshold is set for each simulation case to maximize flow throughputs. Because the antenna gain patterns of the used single-switched-beam antenna and the omnidirectional antenna greatly differ, the transmit powers of the radios used in the MTD and STD schemes were set to different values to maximize their respective network performances. The transmit power of the radio using the single-switched-beam antenna was set to 22 dBm while that of the radio using an omnidirectional antenna was set to 50 dBm.

The ECC-33 model was originally proposed for modeling the path loss effect over 3.5 GHz omnidirectional radios. For this reason, when single-switched-beam antennas are used in simulations, the transmit antenna gain and the receive antenna gain used in this model will be recomputed using the gain pattern shown in Fig. 4.13 and the relative angle between the orientations of the antennas of the transmitting and receiving nodes.

4.3.2 Performance Metrics

Two performance metrics are used to study the average throughput performances of the application flows in simulations: 1) the Average Throughput of a UDP Flow in a network (ATUF) and 2) the Average Throughput of a TCP Flow in a network (ATTF). In addition, the coefficients of variation (CVs) of the ATUF and ATTF values of all flows in a network are presented to study the fluctuation degree of application throughputs under the evaluated schemes. In addition to these application-layer metrics, we use two MAC-layer performance metrics to evaluate the performance gain of the proposed scheme. One is the Average TxOpp Utilization of Nodes (ATOUN) and the other is the Average Three-way Handshake Procedure Time (ATHPT). In the following, we explain the definitions of these performance metrics in detail.

ATUF and ATTF

The ATUF metric is used to estimate the average throughput that a greedy UDP flow can achieve whereas the ATTF metric is used to estimate the throughput that a greedy TCP flow can achieve. The ATUF metric is defined as the average throughput of a UDP flow across all network nodes, which is defined as follows:

$$\text{ATUF} = \frac{\text{TTAF}}{\text{TNST}}, \tag{4.30}$$

where TTAF (total throughputs of all flows) denotes the sum of all nodes' average UDP-flow throughput samples across the different runs of a case, which is defined by the following equation:

$$\text{TTAF} = \sum_{m=1}^{nc} \sum_{i=1}^{nn} \sum_{n=ts\_time+1}^{te\_time-1} udp\_th_{ijm}(n), \forall j \in nbr_1(i), \tag{4.31}$$

where nc denotes the number of runs that a simulation case was performed and nn denotes the number of nodes in the simulated network. The ts_time denotes when a UDP-flow sender program starts transmitting data packets in seconds and the te_time denotes when a UDP-flow sender program stops transmitting data packets in seconds. In our simulations, ts_time and te_time are set to 200 and 499 seconds, respectively. The $udp\_th_{ijm}(n)$ denotes the average throughput of a UDP flow from node i to node j during the n_{th} second in the m_{th} run. The definition of $nbr_1(i)$ has been given in Eq. 2.2.

TNST (total number of sampled throughput [2]) denotes the total number of sampled UDP-flow throughputs across all flows and all runs of a case and is defined as follows:

$$\text{TNST} = \sum_{m=1}^{nc} \sum_{i=1}^{nn} \sum_{n=ts\_time+1}^{te\_time-1} |nbr_1(i)|. \tag{4.32}$$

The definition of ATTF is similar to that of ATUF, except that ATTF uses the following formula to calculate its TTAF:

$$\text{TTAF} = \sum_{m=1}^{nc} \sum_{i=1}^{nn} \sum_{n=ts\_time+1}^{te\_time-1} tcp\_th_{ijm}(n), \forall j \in nbr_1(i), \tag{4.33}$$

where $tcp\_th_{ijm}(n)$ denotes the average throughput of a TCP flow from node i to node j during the n_{th} second in the m_{th} run.

[2] In our simulations, a flow is sampled every one second to record its average throughput in the past one second.

CV-ATUF and CV-ATTF

The coefficients of variation (CV) of ATUF and ATTF values (denoted as CV-ATUF and CV-ATTF) are used to estimate the fluctuation degree of a traffic flow's throughput during simulation. The zero value of CV-ATUF (or CV-ATTF) indicates that the throughput of a greedy UDP (or TCP) flow in a network is stabilized at any given time. In contrast, a large CV-ATUF (CV-ATTF) value means that the throughput of a greedy UDP (TCP) flow in a network fluctuates greatly over time. The definition of CV-ATUF is defined as follows:

$$\text{CV-ATUF} = \frac{\text{StdDev\_ATUF}}{\text{ATUF}}, \tag{4.34}$$

where StdDev_ATUF denotes the standard deviation of all UDP-flow throughput samples in a case, which is defined by the following equation:

$$\text{StdDev\_ATUF} = \sqrt{\frac{\sum_{m=1}^{nc}\sum_{i=1}^{nn}\sum_{n=ts\_time+1}^{te\_time-1}(udp\_th_{ijm}(n) - \text{ATUF})^2, \forall j \in nbr_1(i)}{\text{TNST}}}. \tag{4.35}$$

The definition of CV-ATTF is similar to that of CV-ATUF and is shown as follows:

$$\text{CV-ATTF} = \frac{\text{StdDev\_ATTF}}{\text{ATTF}}, \tag{4.36}$$

where StdDev_ATTF denotes the standard deviation of all TCP-flow throughput samples in a case, which is defined as follows:

$$\text{StdDev\_ATTF} = \sqrt{\frac{\sum_{m=1}^{nc}\sum_{i=1}^{nn}\sum_{n=ts\_time+1}^{te\_time-1}(tcp\_th_{ijm}(n) - \text{ATTF})^2, \forall j \in nbr_1(i)}{\text{TNST}}}. \tag{4.37}$$

ATOUN

The ATOUN metric is used to evaluate whether a network efficiently utilizes the control-plane bandwidth. As explained in Section 2.2, MEA guarantees that on each TxOpp, from the perspective of a node, only one node can transmit an MSH-DSCH message within its two-hop neighborhood. That is, nodes within the same two-hop neighborhood cannot transmit MSH-DSCH messages on the same TxOpp. Therefore, the average TxOpp utilization viewed from a node j using a single-switched-beam antenna can be defined as follows:

$$\mathrm{AvgTxOppUtil}(j) = \frac{(\sum_{k=0}^{NumTD-1} \sum_{i\in(nbr(j_k)-\{j\})} \mathrm{txnum}(i)) + \mathrm{txnum}(j)}{\mathrm{total}(j)}, \tag{4.38}$$

where txnum(i) denotes the number of TxOpps won by a node i; total(j) denotes the number of total TxOpps that have elapsed since node j joins the network; NumTD denotes the number of TDs that node j has; and the definition of $nbr(j_k)$ has been given in Eq. 3.6. On the other hand, the average TxOpp utilization viewed from a node j using an omnidirectional antenna can be defined as:

$$\mathrm{AvgTxOppUtil}(j) = \frac{\sum_{i\in nbr(j)} \mathrm{txnum}(i)}{\mathrm{total}(j)}. \tag{4.39}$$

The AvgTxOppUtil(j) can be used to indicate how well the nodes in node j's two-hop neighborhood together utilize the TxOpps of the network. Ideally, the average TxOpp utilization viewed from each node should be 100%, indicating that, from the perspective of each node, no TxOpps are left unused in its two-hop neighborhood. Recall that on each TxOpp, only one node can transmit its MSH-DSCH message; thus each node has to choose a distinct TxOpp to transmit its MSH-DSCH message on the time axis. We note that due to the conservative eligibility determination rule used in the standard, the maximum value of this metric will not exceed 100%. The ATOUN metric is defined as the average of the *AvgTxOppUtil* values across all nodes in a case. The definition of ATOUN is shown as follows:

$$\mathrm{ATOUN} = \frac{\sum_{j=1}^{m} \mathrm{AvgTxOppUtil}(j)}{m}, \tag{4.40}$$

where m is the number of nodes in a simulation case. Each ATOUN result presented in Section 4.3.3 is the average ATOUN value of a case across ten runs. The ATOUN metric reflects the TxOpp utilization of a network. A higher value of ATOUN indicates that a network case has a higher utilization of TxOpps while a lower value of this metric indicates that a network case has a lower utilization of TxOpps.

ATHPT

The ATHPT metric is defined as the average time required to complete a THP (which establishes a minislot allocation) across all nodes in a case. A node is allowed to transmit data packets to another node only after it has scheduled a minislot allocation with that

Table 4.6: The used holdoff times of node 1's TDs

TD	Average Used Holdoff Time	Standard Deviation
0	5.464	0.9623
1	2.162	2.6064
2	5.465	0.9274
3	5.430	0.9466

node. For this reason, ATHPT significantly influences the packet delay time experienced by upper-layer application programs. The definition of this metric is explained here. For a case, we first average the times required to establish minislot allocations for a node i (denoted as THPT(i)) using the following equation:

$$\text{THPT}(i) = \frac{\sum_{j=1}^{n} \text{t}_{ij}}{n}, \tag{4.41}$$

where t_{ij} denotes the time required to establish the j_{th} minislot allocation of node i with one of its neighboring nodes and n denotes the number of minislot allocations that node i establishes during simulation. We then compute the ATHPT value of a case as follows:

$$\text{ATHPT} = \frac{\sum_{i=1}^{m} \text{THPT}(i)}{m}, \tag{4.42}$$

where m is the number of nodes in a simulation case. Similar to ATOUN, each ATHPT result presented in Section 4.3.3 is the average ATHPT value of a case across ten runs.

4.3.3 Simulation Results

Effects of Our Proposed Randomness Design

We first examine the effects of the randomness design used in TMEA-D on the control message scheduling of each TD. Suppose that the TD 1 of node 1 has data to send and the other TDs of node 1 do not have data to send. We show the the holdoff times used by the TDs of node 1 in Tab. 4.6. One can see that using TMEA-D TDs that have data to send on average can obtain smaller holdoff times to schedule its control message schedulings. In contrast, TDs that do not have data to send are forced to use larger holdoff times to schedule their control message schedulings. These results show that our proposed TMEA-D algorithm can make active TDs use smaller holdoff times to decrease the time for completing their THPs.

Effects of Holdoff Exponent Values

In this section, we studied the effects of nodes' holdoff times on network performances. For the STD scheme, a given holdoff time exponent value x means that all nodes use the 2^x TxOpps as their holdoff times in simulations, while for the MTD scheme a given holdoff time exponent value x means that the S_{active} set used in the TMEA-D algorithm is $\{x$, x+1, x+2, .., min(x+ Num_of_ActTDs -1, max_exp)$\}$, where x is from 0 to 4 in our simulations. In this series of simulations, the requested frame duration and the number of requested minislots per frame in a THP were set to 32 frames and 30 minislots, respectively.

Fig. 4.14 shows the ATOUN results of the STD and MTD schemes over different holdoff time exponent values. As one knows, increasing the holdoff time exponent value will increase the transmission intervals of nodes' control messages. Thus, the TxOpp utilization of the network will decrease, when this value increases. One interesting phenomenon is that the TxOpp utilization of the STD scheme is higher than that of the MTD scheme, when the holdoff time exponent value is very small (i.e., below 2), but drops more rapidly than that of the MTD scheme, when this value increases. We explain this phenomenon from two aspects. First, when using the MTD scheme a node i uses multiple MEAIs to manage its transmission domains. Due to this design, from the perspective of node i's neighboring nodes, on TxOpp T the set of TxOpps for which node i will not contend is denoted as $S^i_{inact\_tx}(T)$ and given as follows:

$$S^i_{inact\_tx}(T) := \cap H^i_j(T), 0 \leq j \leq \text{Num\_of\_ActTDs - 1}, \tag{4.43}$$

where $H^i_j(T)$ denotes the holdoff time interval of node i's MEAI j known by node i's neighboring nodes on TxOpp T. For node i, it is impossible that $H^i_j(T)$, $\forall$ existing MEAI j, exactly overlap. Thus, the size of $S^i_{inact\_tx}(T)$ in the MTD scheme is much less than that in the STD scheme. This means that nodes in the MTD scheme more conservatively choose TxOpps to transmit control messages to avoid inter-node scheduling conflicts. As a result, when the flexibility of TxOpp scheduling is large (e.g. all nodes can use small holdoff time exponent values), the TxOpp utilization of the STD scheme can be better than that of the MTD scheme. Second, the reason why the TxOpp utilization of the MTD scheme decreases slower than that of the STD scheme, as the holdoff time exponent value increases, is explained here. The MTD scheme employs multiple MEAIs to manage directional control message transmissions and all of these MEAIs need to find a conflict-free TxOpp to transmit their control messages in their respective TDs. Nodes using the MTD scheme therefore needs to consume more TxOpps than the STD scheme. As a result, when nodes' holdoff time intervals becomes large, nodes in the MTD scheme will use more TxOpps than those in the STD scheme, which makes the TxOpp utilization of the MTD scheme drops more slowly than that of the STD scheme.

However, as can be seen in Fig. 4.15, the time required for a node to establish a minislot allocation is insensitive to the holdoff time exponent value when it is below 3. This is because the procedure to establish a minislot allocation is three-way based, which should finish a "requester-granter-requester" control message transmission sequence. Due to the randomness of the distributed TxOpp scheduling used in the 802.16(d) mesh CDS mode, the requesting and granting nodes may not always achieve the most efficient TxOpp scheduling to minimize the time for establishing a minislot allocation. In addition, although decreasing the holdoff time exponent value increases the TxOpp scheduling flexibility of nodes, it does not guarantee that a node that is performing a THP can always win a smaller TxOpp. (This is affected by how many MEAIs on this node and its neighboring nodes are performing THPs at the same time.) For these reasons, even when the holdoff time exponent value is set to very small (e.g. no more than 3), the average of the time required for establishing a minislot allocation remains the same.

As one also sees, a larger holdoff time exponent value (e.g., 4) can increase the time required for establishing a minislot allocation. These results confirm the findings presented in [2][3][4][21]. Another noticeable phenomenon is that the average minislot allocation establishment time of a node using the MTD scheme is greatly higher than that of a node using the STD scheme. Such a phenomenon results from two reasons. One is that, due to the directivity of a single-switched-beam antenna, a node using the MTD scheme cannot exchange control messages with a specific peer node on every TxOpp that it wins. Instead, it is only allowed to communicate with a specific peer node on TxOpps won by its MEAI that manages the TD where this peer node resides. In contrast, nodes using the STD scheme can communicate with any of its neighboring nodes on every TxOpp that it wins. Due to this difference, nodes using the MTD scheme require more time to complete a THP and obtain a minislot allocation.

However, because the MTD scheme is capable of utilizing the spatial reuse advantage

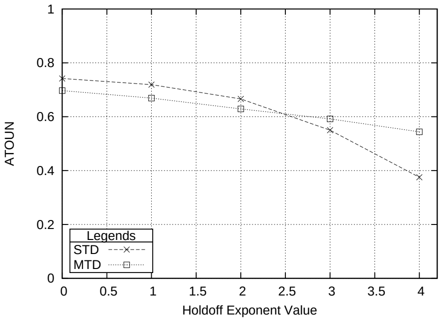


Figure 4.14: ATOUN results over different holdoff time exponent values

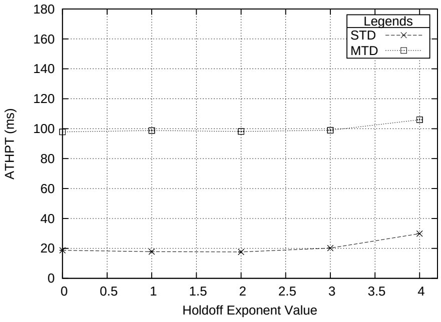


Figure 4.15: ATHPT results over different holdoff time exponent values

of switched-beam antennas, it still can outperform the STD scheme on UDP and TCP flow throughputs. As shown in Figures 4.16 and 4.17, the MTD scheme can on average outperform the STD scheme on UDP flow throughputs by a factor of 2.71 and on TCP flow throughputs by a factor of 5.88, regardless of the used holdoff time exponent value. The reason why TCP performs worse than UDP on average throughput is that TCP uses a complicated congestion control algorithm to prevent network bandwidth from being exhausted by a single flow, which usually regards packet losses as an indication of network congestion. Because the IEEE 802.16(d) mesh CDS mode schedules minislots in a distributed manner, the time for a node to obtain a minislot allocation may fluctuate and the number of minislots obtained in a minislot allocation may greatly vary. In this condition, an outgoing network interface needs to temporarily store packets in its own packet queue. If the packet queue of an interface becomes full, packets sent from upper-layer applications will be dropped, which may make TCP unnecessarily reduce its congestion window size and under-utilize link bandwidth.

In this section, we showed that the holdoff time exponent value has great impacts on TxOpp utilization. However, because the IEEE 802.16(d) mesh CDS mode uses a distributed three-way handshake design to schedule minislot allocations on the data plane, as long as the used holdoff time exponent value is not too large (e.g. above 4), the average time for nodes to negotiate a minislot allocation is insensitive to the holdoff time exponent value. Thus, the average UDP and TCP flow throughput results of both the MTD and STD schemes are unchanged when the holdoff time exponent value is between 0 and 4.

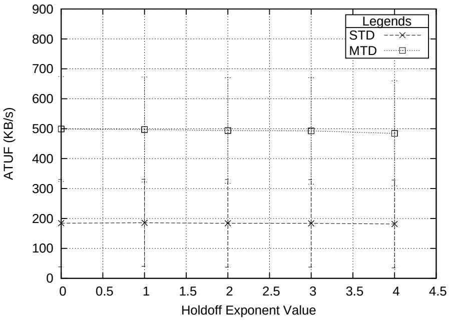


Figure 4.16: ATUF results over different holdoff time exponent values

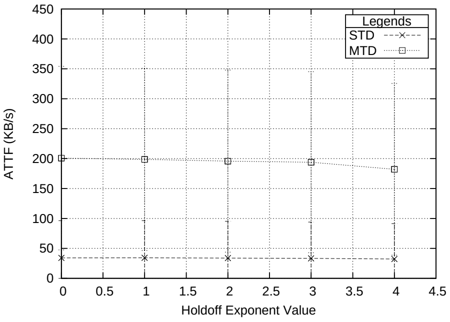


Figure 4.17: ATTF results over different holdoff time exponent values

Effects of Requested Frame Duration per THP

In this section, we studied whether the frame duration of a minislot allocation affects the network performances under the MTD and STD schemes. In this series of simulations, the holdoff time exponent value was set to 0 for both the MTD and STD schemes. The number of requested minislots per frame in a THP was set to 30 minislots. The requested frame duration for a minislot allocation was set to 2^0, 2^1, 2^2, 2^3, 2^5, and 2^7, respectively, which are all of the values allowed in the standard.

Fig. 4.18 and Fig. 4.19 show the average UDP flow throughput and average TCP flow throughput results over different requested frame duration in a THP, respectively. One intuitive result is that increasing the requested frame duration in a THP can increase the utilization of minislots, which results in increased UDP and TCP flow throughputs for both of the evaluated schemes.

A noticeable phenomenon is that, when the requested frame duration per THP is below 2^3 frames, the UDP and TCP flow throughputs achieved by the MTD scheme is only the same as those achieved by the STD scheme. This is because, as discussed previously, using the MTD scheme nodes on average need 100 ms (i.e., 10 MAC-layer frames) to obtain a minislot allocation. To prevent a node from monopolizing link bandwidth, in our implementation a node A will be triggered to perform a THP with a neighboring node B, only when 1) it has data destined to node B and 2) node A does not possess any valid minislot allocation granted by node B. Due to this design and the long ATHPT property of the MTD scheme, if the requested frame duration in each THP does not exceed $\lceil(\text{ATHPT}/\text{Num\_of\_ActTDs})\rceil$ frames, nodes using the MTD scheme will not be able to schedule minislots as tight as those using the STD scheme. In contrast, when

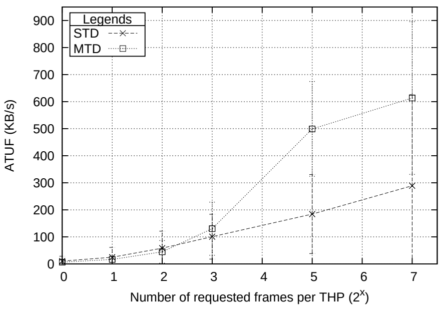


Figure 4.18: ATUF results over different requested frame durations in a THP

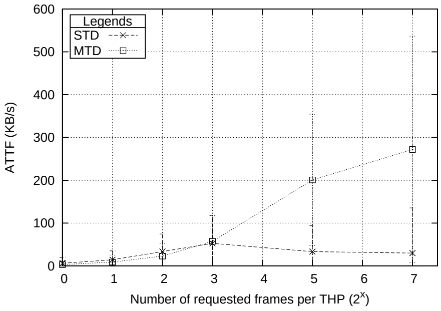


Figure 4.19: ATTF results over different requested frame durations in a THP

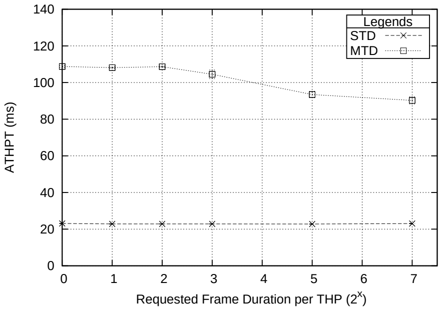


Figure 4.20: ATHPT results over different frame durations under the UDP flow case

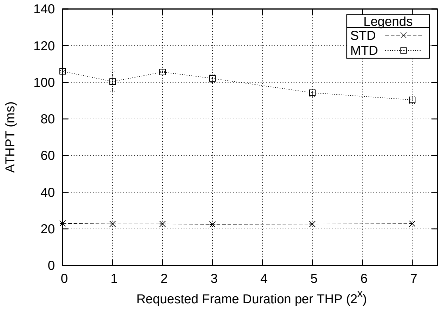


Figure 4.21: ATHPT results over different frame durations under the TCP flow case

the requested frame duration greatly exceeds $\lceil(\text{ATHPT}/\text{Num\_of\_ActTDs})\rceil$ frames, due to the spatial reuse advantage, the MTD scheme can greatly outperform the STD scheme on UDP and TCP flow throughput performances.

We plotted the ATHPT results of the two evaluated schemes under the UDP and TCP traffic cases over different requested frame durations in Fig. 4.20 and Fig. 4.21. The results show that the time required for a node to complete a THP is less related to the requested frame durations.

Effects of Numbers of Requested Minislots per THP

In this section, we studied the effects of requested minislots in a THP (different MAC-layer traffic loads) under the MTD and STD schemes. In this series of simulations, the holdoff time exponent value was set to 0 and the requested frame duration in a THP was set to 2^7 frames. The number of requested minislots per frame in a THP was set to 0, 10, ..., 70 minislots, respectively.

The average UDP flow throughput results over different numbers of requested minislots per THP is plotted in Fig. 4.22, which show that the achieved throughput of UDP flows

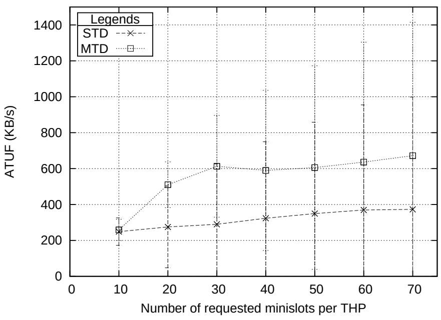


Figure 4.22: ATUF results over different numbers of requested minislots in a THP

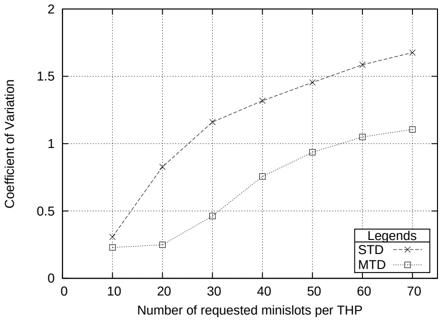


Figure 4.23: CV-ATUF results over different numbers of requested minislots in a THP

under the MTD scheme significantly outperforms that of UDP flows under the STD scheme over most of the evaluated MAC-layer traffic loads. These results evidence that the MTD scheme can effectively exploits spatial-reuse advantages of single-switched-beam antennas to provide more network capacity for applications. One may notice that, when the number of requested minislots per THP is 10 only, the average UDP throughput results of the two evaluated schemes are close. This is because, when the generated traffic load is very light, the STD scheme can accommodate it without causing network congestion. However, when the number of requested minislots in a THP increases (i.e., the generated MAC-layer traffic load increases), the STD scheme quickly reaches its saturation point and cannot keep up with the performance of the MTD scheme.

Fig. 4.23 shows the CV-ATUF results over different numbers of requested minislots in THPs, which indicate the fluctuation degree of the achieved throughputs of flows in a network over time and the fairness of network bandwidth allocation among them. One can see that the CV-ATUF values of the MTD scheme are much lower than those of the STD scheme, showing that network applications can achieve a more stable throughput over time under the MTD scheme than under the STD scheme. These results also indicate that network bandwidth sharing among these competing UDP flows is fairer under the MTD scheme.

Fig. 4.24 shows the ATTF results of the two evaluated schemes over different numbers of requested minislots per THP. There are several findings about this figure. First, the MTD scheme greatly outperforms the STD scheme over different traffic loads. Second, when the number of requested minislot is larger than 20 minislots, the average TCP flow throughputs achieved by the two evaluated schemes start to decrease. These results are explained here. When the number of requested minislots in each THP increases, the

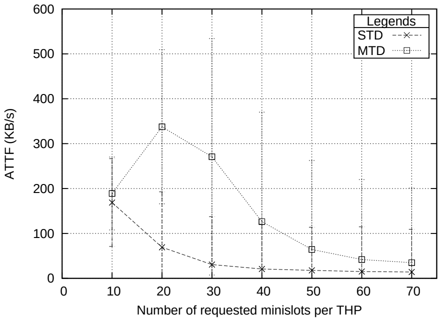


Figure 4.24: ATTF results over different numbers of requested minislots in a THP

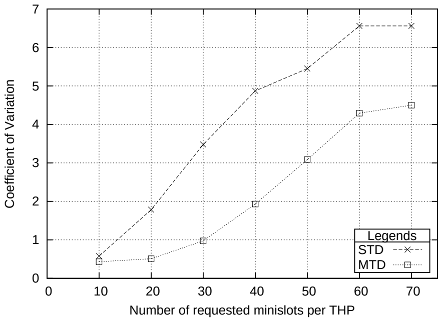


Figure 4.25: CV-ATTF results over different numbers of requested minislots in a THP

number of minislots that each node can obtain in a THP will drastically fluctuate. As explained in Section 4.3.3, when outgoing packets cannot be sent immediately, they will be stored in the packet output queue of the interface card. However, TCP speculates about the amount of data that it can transmit mainly based on detected packet losses and therefore it does not take the number of minislots obtained by the MAC-layer in each THP into account. For this reason, TCP may inject an excessive number of packets down to the MAC-layer and thus may generate packet dropping at the MAC-layer. Such packet dropping will trigger TCP's congestion control mechanism to reduce its transmission speed.

Although TCP is more sensitive to network congestion, as shown in Fig. 4.25, the MTD scheme still significantly outperforms the STD scheme on CV-ATTF, indicating that a TCP flow under the MTD scheme will achieve a more stable throughput over time than that under the STD scheme. These results also indicate that these competing TCP flows share the network bandwidth more fairly under the MTD scheme.

We finally studied the ATOUN and ATHPT results of the MTD and STD schemes under different numbers of minislots requested in THPs. The results are as expected. The ATOUN metric measures the utilization of TxOpps, which are on the control plane. However, varying the number of minislots requested in THPs only affects how minislots on the data plane will be allocated. Since TxOpps and minislots are not directly related, changing this variable has minor effects on the TxOpps utilization. In addition, varying the MAC-layer traffic load in each THP does not change the frequency that a node performs a THP. Varying this variable, therefore, has minor effects on the time required for finishing a THP.

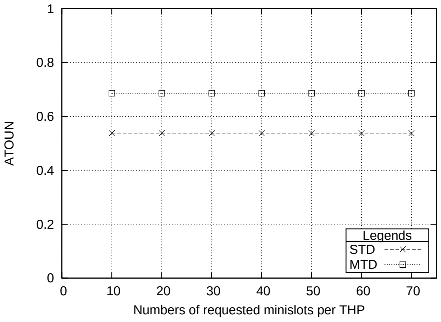


Figure 4.26: ATOUN results over different numbers of requested minislots in a THP

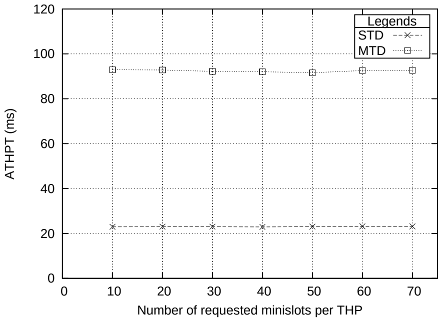


Figure 4.27: ATHPT results over different numbers of requested minislots in a THP

Effects of Mixed Traffic

In this section, we studied the impacts of the MTD scheme on the network quality experienced by different traffic flow types. In this series of simulations, node 1 establishes traffic flows with different types to each of its neighboring nodes. The detailed setting is explained here: node 1 establishes 1) a TCP flow to node 9; 2) a UDP flow that transmits a 1400-byte packet per 0.1 second to node 13; 3) a UDP flow that transmits 800-byte packets using an exponential inter-arrival time distribution with the mean value 1.0 second, the minimum value 0.1 second, and the maximum value 2.0 second; and 4) a UDP flow that transmits 1400-byte packets with the inter-arrival time set to a normal distribution (the minimum value set to 0.1 second and the maximum value set to 2.0 second). The other nodes in the network runs greedy UDP flows with each of their neighboring nodes to generate background traffic.

The main effect of the MTD scheme on traffic flows is the increase of packet delays. We show the average packet delay results of different flows on node 1 in Tab. 4.7. Each presented result is the average across the packet delays of all packets received by the flow. One can find that, due to the time overheads of transmitting control messages, the MTD scheme greatly increase the packet delays of traffic flows. One first sees that the UDP flow with the constant inter-arrival time distribution experiences the shortest packet delays. This is because nodes in the IEEE 802.16 mesh CDS-mode network transmits data in a reservation-based manner. Thus, when the packet inter-arrival time of a traffic flow is smaller, more packets can be transmitted over the same minislot allocation. As a result, the packet delays experienced by the traffic flow can be reduced.

In contrast, when the packet inter-arrival time is higher (e.g., the UDP flows with the

Table 4.7: Experienced packet delays of different flows

Flow Type	Average Packet Delay (ms)
UDP flow with a constant inter-arrival time	73.92
UDP flow with an exponential inter-arrival time	218.20
UDP flow with a normal inter-arrival time	198.80

Table 4.8: Throughputs obtained by different flows

Flow Type	Average Flow Throughput (KB/s)
TCP flow	0.003
UDP flow with a constant inter-arrival time	13.556
UDP flow with an exponential inter-arrival time	1.019
UDP flow with a normal inter-arrival time	1.27

exponential and uniform inter-arrival time distributions used in our simulations), their transmitted packets may not be transmitted over the same minislot allocation. In this condition, the packet delays experienced by traffic flows are increased. To reduce the packet delays experienced by applications, each node can in advance trigger its THPs as long as the output buffers of its connections have data to send. This design is an item of our future work and not included in our current implementation.

We also presented the average throughput obtained by each flow in Tab. 4.8. One can see that the throughputs of UDP flows greatly match those derived from their inter-arrival time distributions. The reason why the throughputs of the TCP flow are down to nearly zero are explained here. The ACK packets of the TCP flow should compete the slots of node 9's outgoing interface queue with a greedy UDP flow. In this condition, most ACK packets can be dropped at node 9's output interface queue. This will cause the TCP flow to unnecessarily trigger its congestion control mechanism and eventually fully throttle down its packet transmissions. As a result, the throughputs obtained by the TCP flow is very low. This problem can be solved by separating the outgoing interface queues of the TCP packets from those of the UDP packets. This mechanism, however, is an item of our future work and not included in our current implementation.

Effects of Rayleigh Fading

In this section, we finally studied the throughput performances of the MTD and STD schemes when Rayleigh fading is present. The presence of Rayleigh fading means that 1) there is no line-of-sight path for a pair of transmitting and receiving nodes and 2) there are some objects moving in the network. Such a scenario is possible to occur in metropolitan areas where network nodes are surrounded by high buildings and many vehicles and pedestrians move at different speeds. Fig. 4.28 shows the relationship between the average UDP flow results and the value of Rayleigh fading variance. A larger value of the Rayleigh fading variance means a higher fluctuation level of received signal power on a receiving node while a smaller value of the Rayleigh fading variance means a lower fluctuation level of received signal power on a receiving node.

By observing the results shown in Fig. 4.28, several findings are drawn. First, it is as expected that, when the fading variance value increases, the throughput performances of both the MTD scheme and the STD scheme drastically decrease. This is because, when the fading variance value is large, the power of received signal greatly fluctuates. On one hand, the reduction of received signal power due to Rayleigh fading increases the number of bit errors in received packets, which significantly decreases the goodputs obtained by nodes. On the other hand, the fluctuation of received signal power due to Rayleigh fading increases the signal interference level among neighboring nodes. In this condition, two bad effects results. One is that the Signal to Interference and Noise Ratio (SINR) values will be greatly decreased on receiving nodes, which also increases the number of bit errors in received packets. The other is that the number of control message collisions and data packet collisions will be increased due to undesired signal interference.

Although wireless transmission is difficult over a Rayleigh fading channel, one can see that nodes using the MTD scheme is superior than those using the STD scheme in resisting the signal strength fluctuation resulting from the Rayleigh fading effect. This phenomenon can be explained from two aspects. First, a single-switched-beam antenna is capable of focusing its gain on only several directions. Thus, a node using a single-switched-beam antenna will not leak too much signal power in the directions that are not covered by the main lobe and the side lobes of its antenna. Second, due to the use of TMEA-D, nodes using the MTD scheme can know when and to which directions they should point their antennas to receive control messages and data. Such knowledge allows receiving nodes to achieve the largest receive antenna gain in the direction where the transmitting node is.

In addition, although the MTD scheme uses multiple MEAIs to schedule transmissions of control messages, when scheduling these transmissions, the core algorithm TMEA-D used in these MEAIs will consider the contention of all neighboring nodes in $\cup nbr(i_m), \forall$ TD $m \in$ node i. Thus, using the MTD scheme when node i transmits an MSH-DSCH/MSH-NCFG message to node j, nodes that are two-hop away from nodes i and j can avoid pointing their own antennas to these two nodes, which prevents themselves from being greatly interfered by node i's transmission. In contrast, due to the omnidirectional transmission and reception property, nodes using omnidirectional antennas (such as those in the STD scheme) cannot alleviate the signal interference resulting from the Rayleigh fading effect.

Such signal inference on MSH-DSCH and MSH-NCFG message transmissions generates more MSH-DSCH and MSH-NCFG message collisions, which results in two bad effects. One is that it is more difficult for nodes using the STD scheme to complete THPs (and thus obtain minislot allocations to transmit data). The other is that it is more difficult for nodes using the STD scheme to join the network because in this bad channel condition successfully receiving an MSH-NCFG message (which is necessary for the network entry process of a node) is more difficult for nodes using the STD scheme.

For the above reasons, nodes using the MTD scheme have higher probability to complete THPs (which are used to schedule data transmissions) than those using the STD scheme, when the Rayleigh fading is present. Thus, the MTD scheme has higher efficiency on performing THPs than the STD scheme. In addition, nodes using the MTD scheme can successfully join the network, even when the Rayleigh fading is present in the network. In contrast, in a network using the STD scheme several nodes may fail to join the network. As a result, the throughput performances achieved by nodes using the MTD scheme are greatly better than those achieved by nodes using the STD scheme over wireless channels with the presence of the Rayleigh fading.

Fig. 4.29 shows the TCP flow throughput results of the two evaluated schemes over different Rayleigh fading variance values. One noticeable phenomenon is that, when the Rayleigh fading variance value increases, the average flow throughput achieved by TCP decreases more sharply than that achieved by UDP. This is because on a Rayleigh fading channel with a larger variance value the transmissions of MSH-DSCH messages are no longer guaranteed to be collision-free. In this condition, a THP may fail and thus the time required for a transmitting node to obtain a minislot allocation to transmit data may greatly vary over time. As explained previously, such fluctuation of the available link

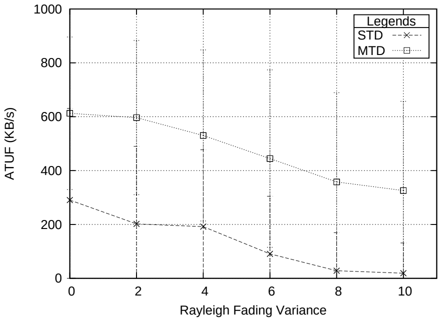


Figure 4.28: The effects of the fading variance on UDP flow throughputs

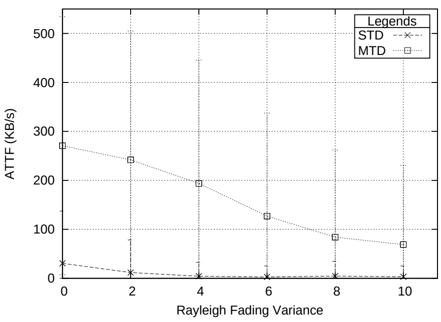


Figure 4.29: The effects of the fading variance on TCP flow throughputs

bandwidth may make TCP excessively transmit packets down to the MAC-layer, which generates undesired packet losses on the local interface. Upon detecting such packet losses, TCP will decrease its transmission speed and thus under-utilize the link bandwidth.

4.3.4 Summary

In this paper, we identify and address the important issues when equipping each node with a single-switched-beam antenna in an IEEE 802.16(d) mesh CDS-mode network. We propose a complete scheme to solve these issues and improve the performances of such networks. This scheme, called the MTD scheme, is implemented on the NCTUns network simulator. Its performances are extensively studied and compared against those of the original standard. Our simulation results show that, because of exploiting the spatial-reuse advantages of single-switched-beam antennas, the proposed MTD scheme significantly outperforms the original STD scheme and greatly increases the network capacity of an IEEE 802.16(d) mesh CDS-mode network. Our simulation results also show that competing TCP and UDP flows achieve more stable throughputs over time and share network bandwidth more fairly under the MTD scheme.

Chapter 5

Discussion

In this chapter, we discuss several aspects that can affect the performance of an IEEE 802.16 mesh CDS-mode network. In Section 5.1, we first point out the problems of initializing the IEEE 802.16 mesh CDS-mode network in general topologies and propose two schemes to alleviate the problems. In Section 5.2, we further propose a two-phase holdoff time scheme to ensure that the initialization of the IEEE 802.16 mesh CDS-mode network in general topologies always succeeds. In Section 5.3, we point out the design flaw of THP in QoS support and propose a solution to enhance the QoS support of THP. Finally, in Section 5.4 we study the effect of the collaborative network-layer routing protocol on WMNs.

5.1 Enhancement of the Network Initialization Process in the IEEE 802.16 Mesh CDS Mode

In an IEEE 802.16 (WiMAX) [1] mesh network, a new network node needs to complete two processes before it can use network resources. First, it should finish the network entry (NENT) process to attach to the network. Second, it should use the link establishment (LinkEst) process to establish links with its neighboring nodes. Only after successfully finishing the two processes, will a new node become a functional node in the network.

Fig. 5.1 shows the procedures of the NENT process of the mesh mode. First, a new node has to listen to Mesh Network Configuration (MSH-NCFG) messages from its neighboring functional nodes to perform coarse time synchronization and obtain necessary system parameters. Second, it should continually monitor MSH-NCFG messages from all of its neighboring nodes and build a physical neighbor list based on the received messages until it receives a MSH-NCFG message from the same node twice [1]. Next, the new node should choose a candidate sponsoring (CS) node from the list and then synchronizes its time with the time of the CS node assuming zero signal propagation delay. The new node then has to contend for a network entry transmission opportunity to send the CS node a Mesh Network Entry message. This message contains a Network Entry Request information element (MSH-NENT:NetEntryRequest) carrying the Node ID of the chosen CS node.

Upon receiving this request message, the CS node either accepts or rejects it. If the CS node cannot service this request, it will send a MSH-NCFG message with Network Entry Reject information element (MSH-NENT:NetEntryRequest) back to the requesting node. If the CS node accepts the request, it will send a MSH-NCFG message with a Network Entry Open information element (MSH-NCFG:NetEntryOpen) to the requesting node and become the sponsor node for this requesting node.

The MSH-NCFG:NetEntryOpen message contains an estimated delay field to help the new node to perform fine-grained time synchronization. It also contains a temporary transmission schedule for the new node to transmit its control messages to the BS node in the network. After receiving the MSH-NCFG:NetEntryOpen message, the new node should transmit a MSH-NENT message with Network Entry Ack information element (MSH-NENT:NetEntryAck) to the sponsor node and start to proceed the basic capability negotiation procedure, the authorization procedure, and the registration procedure using the schedule provided by the sponsor node.

Once the new node finishes the above procedures, it should transmit a MSH-NENT message with Network Entry Close information element (MSH-NENT:NetEntryClose) to notify the sponsor node that it has already finished the necessary procedures and the temporary transmission schedule is no longer needed.

Upon receiving the MSH-NENT:NetEntryClose message, the sponsor node will cancel the temporary transmission schedule dedicated to the new node. Finally, the sponsor node transmits a MSH-NCFG message with Network Entry Ack information element (MSH-NCFG:NetEntryAck) to the new node to terminate this sponsorship. When the new node receives this acknowledgment message, it will become a functional node in the network.

After finishing the NENT process, the new node will have attached to the network.

However, before it can transmit any data packet, it has to establish links with its neighboring nodes. This is accomplished by the LinkEst process, which is a secure three-way handshake procedure used to establish two one-way links between a pair of neighboring nodes.

For example, suppose that node 1 and 2 are neighboring nodes to each other. If node 1 first initiates the link establishment process, it will send a challenge message to node 2. Upon receiving the challenge message, node 2 first authenticates node 1. If the authentication is successful, node 2 will create a link from itself to node 1 and then send back a challenge response message to node 1 as an acknowledgment. Similarly, on receiving the challenge response message of node 2, node 1 first performs authentication for node 2. If the authentication is successful, it will create a link from itself to node 2 and transmit an acceptance message to node 2 to complete the link establishment process.

The performances of the NENT and LinkEst processes are important to the operations of IEEE 802.16 mesh networks because network nodes in such networks are allowed to use network resources only when they can finish these two processes. In this section, we first identify that the NENT process is prone to fail in dense networks. Then, we propose a solution to remedy this problem. According to our simulation results, our proposed solution can increase the success rate of the NENT process in dense networks to 93%. In contrast, the success rate is only about 70% in dense networks when the original process defined in the standard is used. Second, we point out the disadvantages of the LinkEst process defined in the current standard and propose an extension to this process. According to our simulation results, on average, our proposed extension can speed up the LinkEst process by 2.57 times.

5.1.1 The Refined Network Entry Process

As shown in Fig. 5.1, the network entry process consists of six steps and several procedures, briefly described in Section 1. In the NENT process, there are three messages that should be acknowledged — MSH-NENT:NetEntryRequest, MSH-NCFG:NetEntryOpen, and MSH-NENT:NetEntryClose. Each time when such a message is going to be transmitted, an associated timer called T25 will be initiated. When T25 has expired and the corresponding acknowledgment message has not yet been received, a retransmission of the control message will be triggered.

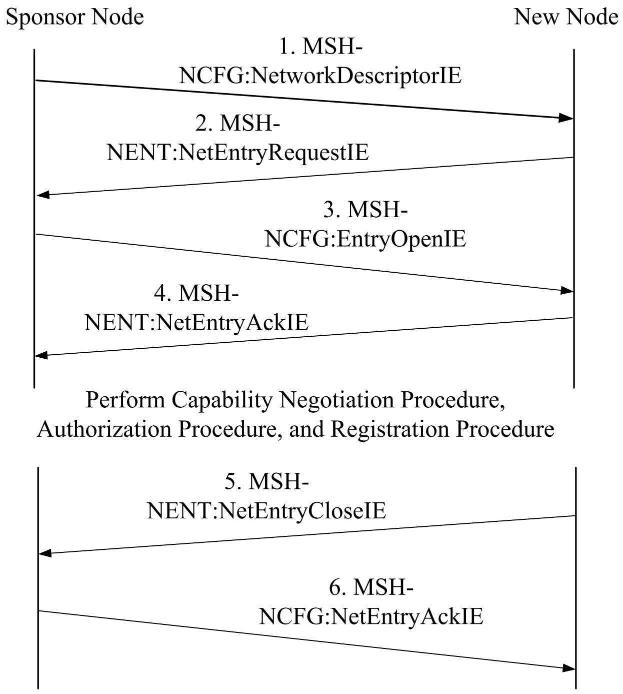

Sponsor Node
New Node
1. MSH-NCFG:NetworkDescriptorIE
2. MSH-NENT:NetEntryRequestIE
3. MSH-NCFG:EntryOpenIE
4. MSH-NENT:NetEntryAckIE
Perform Capability Negotiation Procedure, Authorization Procedure, and Registration Procedure
5. MSH-NENT:NetEntryCloseIE
6. MSH-NCFG:NetEntryAckIE

Figure 5.1: The procedure of the network entry process

The key to the success of the network entry process is successfully exchanging messages between a new node and its sponsor node. However, based on our study, we found a situation in which a sponsor node's MSH-NCFG messages destined to a new node are always collided with the MSH-NCFG messages issued by other functional nodes. As a result, the new node cannot attach to the network and its network initialization process fails.

For a functional node, its transmission times of MSH-NCFG messages are resolved by the distributed pseudo-random algorithm. The algorithm requires the newest information from all functional nodes within its two-hop neighborhood and these information are carried by their MSH-NCFG messages. Thus, when a functional node is going to send a MSH-NCFG message, it should compute the next transmission time of its MSH-NCFG message and put this information into the MSH-NCFG message to be sent.

In a stable network, i.e., all existing nodes are functional and there are no new nodes going to enter this network, the arbitration of control message transmissions using the algorithm defined in the current standard works well. However, when a network is unstable, e.g., there is a new node trying to attach to the network, the well-known *"hidden terminal"* problem may occur. This is because the neighborship of the new node's neighboring

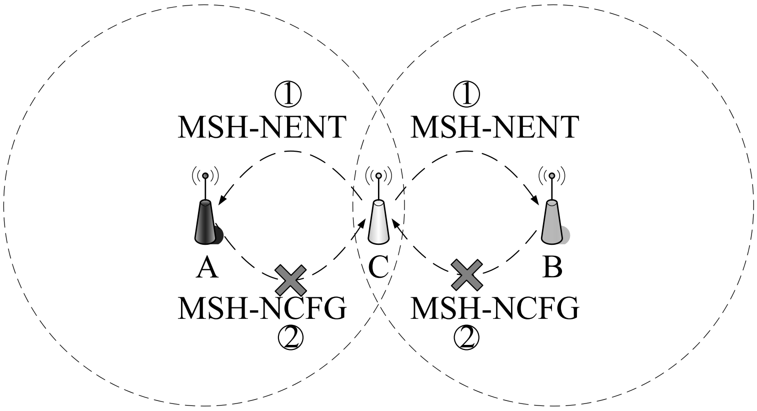


Figure 5.2: A case in which the network entry process fails

functional nodes may get changed.

Fig. 5.2 shows a typical case where the *"hidden terminal"* problem will occur. There is a new node C trying to attach to this network, and nodes A and B are its neighboring functional nodes. The signal coverage of nodes A and B are illustrated by the dotted circles, respectively. In this case, we can see that nodes A and B cannot sense the existence of each other before node C attaches to this network. In such a situation, nodes A and B can transmit packets without the need to coordinate with each other. This is because since simultaneous transmissions from nodes A and B will not result in any collisions before node C is functional.

However, when node C tries to enter this network, nodes A and B will become node C's one-hop neighboring nodes. This means that nodes A and B are no longer allowed to arbitrarily transmit their packets because if they send packets to node C simultaneously, these packets will be collided on node C.

Unfortunately, based on the current standard, node A is unaware of the existence of its potential two-hop neighboring nodes, e.g., node B in this case, until it receives node C's MSH-NCFG messages containing the information about node B. More specifically, nodes A and B will not know each other until node C is functional. Due to this reason, nodes A and B will not coordinate their control message transmissions with each other, and hence successful message reception at node C cannot be guaranteed.

In such a situation, the network entry process of node C will be time-consuming because the control messages sent from its sponsor node, assuming that it chooses node A as its CS node, are very likely to be collided with messages sent by node B. Due to the T25 timeout design defined in the current standard, its sponsor node will retransmit many control messages and thus waste network bandwidth.

In a sparse network, retransmissions of lost control messages may work well because the number of functional nodes contending for one transmission opportunity is small. Nevertheless, retransmissions may not solve the same problem in dense networks where collisions are very likely to occur without elaborate coordination.

If message collisions cannot be reduced significantly in dense networks, it is very possible that a retransmitted message gets collided again. The reason is that for each transmission opportunity there is always at least one functional node contending for its use, and the distributed pseudo-random election algorithm guarantees that a node will win the access to a transmission opportunity if at least one contending node exists.

Our design is to let functional nodes be aware of changes to their neighborship as soon as possible with minimum changes to the current IEEE 802.16 standard. In our proposal, new nodes and sponsor nodes perform the same procedures defined in the original standard while at the same time we force a new node's neighboring functional nodes to be involved in the new node's network entry process.

The refined network entry process is defined as follows. First, a new node transmits a MSH-NENT:NetEntryRequest message containing the node ID of its chosen CS node. Next, the sponsor node either accepts or rejects this request and performs the same procedure as defined in the standard. At the same time, on receiving the MSH-NENT:NetEntryRequest message sent by the new node, functional nodes close to the new node are able to know the existence of the sponsor node by the *”Sponsor Node ID”* field in the MSH-NENT:NetEntryRequest message. Since these neighboring nodes cannot receive the sponsor node's MSH-NCFG messages directly, they do not have any schedule information of the sponsor node. In such a condition, these nodes should treat the sponsor node as a functional node with unknown schedule status, i.e., they should consider that the sponsor node may contend for all following transmission opportunities until it wins one.

Moreover, besides knowing the sponsor node's existence, the neighboring nodes should immediately re-perform the distributed pseudo-random election algorithm considering the existence of the sponsor node. Using this scheme, although the sponsor node itself does not know the existence of these potential two-hop neighboring nodes, its MSH-NCFG messages sent to the new node will not be collided any more with the messages sent by these two-hop neighboring nodes. The reason is that since these neighboring nodes conservatively consider that the sponsor node will contend for all subsequent transmission opportunities, when the sponsor node wins a transmission opportunity using its own two-hop neighborhood information, other nodes close to the new node will also know that the sponsor node wins the transmission opportunity and thus keep silent on that transmission opportunity.

Our proposed scheme eliminates the collisions of control messages sent from the sponsor node to the new node. However, it cannot explicitly prevent collisions of packets whose sources are other functional nodes because only the sponsor node's identity is widely known by functional nodes close to the new node. It is possible that there is other potential neighborship unknown to the new node's neighboring functional nodes. Although our scheme cannot prevent all types of packet collisions, it is sufficient to guarantee successful reception of MSH-NCFG messages sent from the sponsor node to the new node. As a result, our scheme can significantly increase the success rate of the network entry process.

5.1.2 The Refined 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. The process is composed of three steps.

First, the initiating node sends a Link Establishment information element with challenge action code (LinkEstIE:Challenge) using MSH-NCFG message to a neighboring node. In this challenge message, an authentication value is included to help the neighboring node authenticate the initiating node. An authentication value consists of a shared secret key, a frame number N, the ID of the initiating node, and the ID of the neighboring node, where N is the number of the frame in which the initiating node receives the last MSH-NCFG message sent by the neighboring node.

Next, upon receiving the challenge message, the neighboring node should send a MSH-NCFG message containing a LinkEstIE with challenge response action code (LinkEstIE:ChallengeResp back to the initiating node to indicate the acceptance of the request if it accepts this challenge message and finishes creating a link from itself to the initiating node. The challenge response message contains an authentication value and a link ID. The former is used by

Generic MAC Header	Mesh Subheader (Xmt Node ID)	Management Message Type	Management Message Payload

Figure 5.3: MAC header

the initiating node to authenticate the neighboring node while and the latter indicates the ID of the established link.

Finally, after receiving the challenge response message, the initiating node should authenticate the neighboring node. If the authentication is successful, the initiating node should create a link from itself to the neighboring node and send a MSH-NCFG message containing a LinkEstIE with accept action code (LinkEstIE:Accept) to the neighboring node. This LinkEstIE:Accept information element indicates that the authentication is successful and the reverse-direction link has been established.

However, since the final step need not perform the authentication procedure, the standard defines that the format of LinkEstIE:Accept contains only a link ID field, which indicates the ID of the established link from the initiating node to the neighboring node. As shown in Fig. 5.3, using such a format, the only way that the neighboring node knows the source of this LinkEstIE:Accept information element is to use the "Xmt Node ID" field in the mesh subheader, which follows the generic MAC header. This implies that a node cannot establish links with multiple neighboring nodes simultaneously.

For example, if a node A tries to establish links with its two neighboring nodes B and C at the same time, nodes B and C will get confused when receiving a LinkEstIE:Accept from node A. To see this, suppose that node A first transmits a LinkEstIE:Accept for node B to report the ID of the link created for node B. On receiving this LinkEstIE:Accept information element, node C will also consider that this link is dedicated to itself. As a result, this ambiguity will make both nodes B and C use incorrect links to transmit their data packets to node A.

If a functional node is forced to establish links with only one neighboring node at a time, the time to establish links with multiple neighboring nodes will increase significantly as the number of its neighboring nodes increases. The reason is that since the LinkEst process cannot be performed in parallel, each MSH-NCFG message contains only one LinkEstIE and a LinkEst process for a pair of nodes consumes at least three MSH-NCFG transmissions. In a dense network, such a process can be very time-consuming because the interval of each node's MSH-NCFG message transmissions will be increased due to the increased number of the contending nodes.

Furthermore, in such a situation a challenge message for establishing a link is easily postponed by the peer node because the peer node may be performing a LinkEst process with another node. In such a case, a careful implementation for the LinkEst process is necessary to avoid deadlock situations because if a functional node is restricted to establish links with one node at a time, and the challenge message is allowed to be queued (processed later) by the peer node, the LinkEst process itself will have the *"mutual exclusion"* and *"hold-and-wait"* properties, which are the two necessary conditions for deadlock situations.

To address these problems, we proposed an extension to the current standard to prevent deadlock situations and increase the efficiency of the LinkEst process. Our extension is to add the authentication field used in LinkEstIE:Challenge and LinkEstIE:ChallengeResponse messages into LinkEstIE:Accept messages. This format expansion makes LinkEstIE:Accept messages self-contained, i.e., a node is capable of understanding the source of a LinkEstIE:Accept message by the added authentication field.

Also, we allow that multiple LinkEst processes can be performed simultaneously. Since the sources of LinkEstIE:Accept information elements can now be differentiated, multiple LinkEst processes can be performed at the same time without confusing involved nodes.

This extension has two advantages. First, since simultaneous link establishment processes are allowed, the *"mutual exclusion"* and *"hold-and-wait"* properties, which are necessary to cause deadlock situations, in the LinkEst process are eliminated. This is because an initiating node's challenge message now will not be postponed by its peer node when the peer node is establishing links with other ones. Also, since an initiating node now need not wait for a challenge response message indefinitely, the *"hold-and-wait"* property is eliminated. Moreover, because the right to establish links with a specific node is no longer an exclusive resource in the LinkEst process, the *"mutual exclusion"* property no longer exists.

The second advantage is that the required time to establish links with multiple neighboring nodes can be reduced greatly because of two reasons. One reason is that since now a MSH-NCFG message can contain multiple information elements for establishing links, each node can establish links with multiple neighboring nodes at the same time. The other reason is that since challenge messages are unlikely to be postponed under this extension, the time delayed by peer nodes can be further reduced.

5.1.3 Performance Evaluation

We used NCTUns [19] to evaluate our work. Over NCTUns, we implemented two MAC modules, simulating BS-related and SS-related functions, respectively, and an OFDM module responsible for simulating the OFDM air interface and channel codec.

Twenty connected network topologies are generated randomly to minimize influences of specific topologies. Ten of them are composed of one hundred nodes scattered within a square area of side length being 2500 meters, and the others are composed of fifty nodes scattered within a 1500x1500 square area. For each topology, we conducted twenty simulation runs using different random number seeds. All simulations were run on three desktops, equipped with a Pentium-4 3.0 GHZ CPU and 2GB main memory, respectively, and the total simulated time for each case is 500 seconds. For simplicity, routing paths are generated statically using Dijkstra's shortest path algorithm.

Fig. 5.4 shows the cumulative distribution function (CDF) of success rates of network entry processes using four scheme combinations, and Fig. 5.5 shows the CDF of times for establishing links with all neighboring nodes using four scheme combinations. The "standard" curve denotes the result using the original processes defined by the standard; the "New NENT" curve denotes the result using our enhanced NENT process and the original LinkEst process; the "New LinkEst" curve represents the result using the original NENT process and our extended LinkEst process, while the "New Entry+LinkEst" curve represents the result using our enhanced NENT process and extended LinkEst process.

Since behaviors of fifty-node cases are similar to those of one hundred-node cases, we only explain results of one hundred-node cases. For cases with 100 nodes, we see that although within the first 180 seconds, the number of cases finishing the NENT process using the enhanced NENT process is less than that using the original one, after 180 seconds the number of finished cases using the enhanced NENT process is more than that using the original one. This is because the enhanced NENT process forces functional nodes close to new nodes to re-compute the next NCFG transmission opportunity and relinquish one NCFG transmission opportunity which a sponsor node wins. Thus, the progress of network initialization is slightly delayed due to delayed transmissions of several MSH-NCFG messages. Despite the slower start, our scheme significantly reduces the number of MSH-NCFG message collisions, and hence the success rate of the NENT process in the long run can be increased.

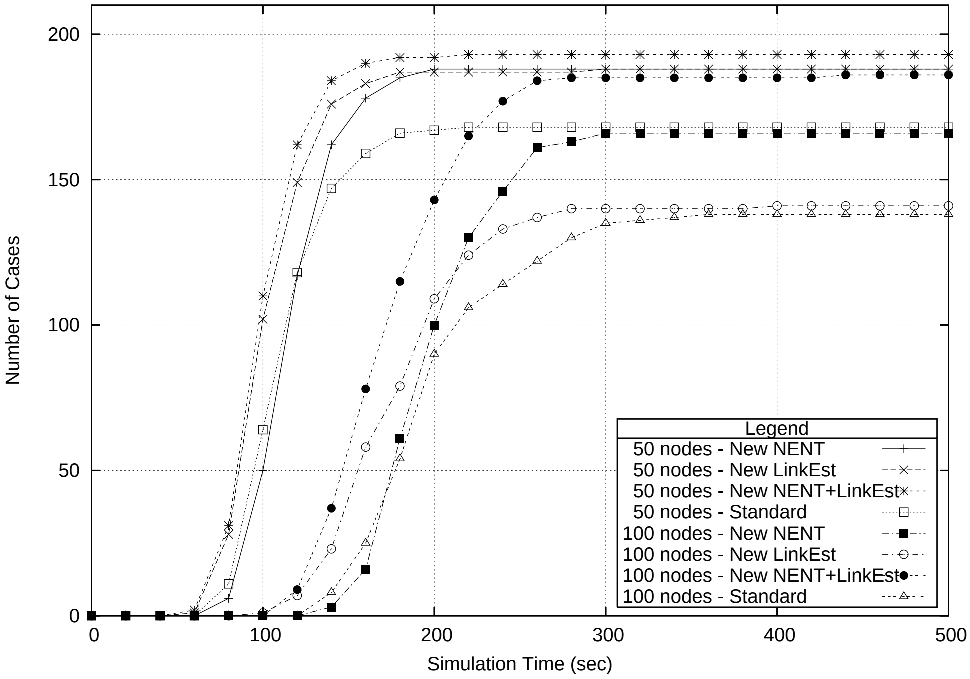


Figure 5.4: The CDF of success rates of network entry processes

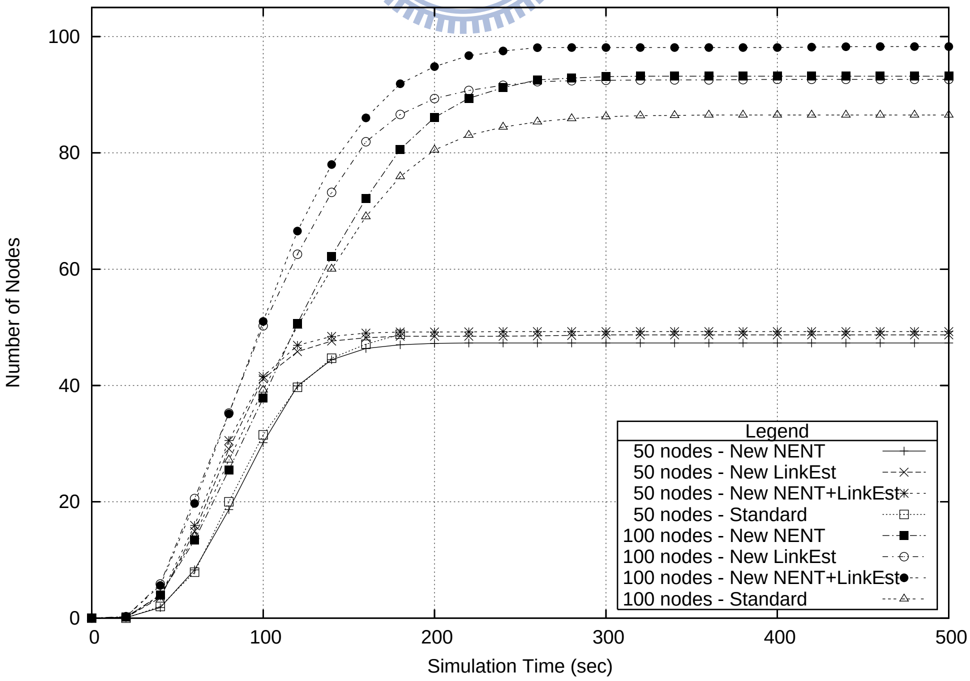


Figure 5.5: The CDF of nodes' required times for establishing links with all neighboring nodes

Table 5.1: Success rates of NENT processes

Scheme\Case	50-node Case	100-node Case
	Success Rate (%)	Success Rate (%)
Standard	84.0	69.0
New NENT	94.0	83.0
New LinkEst	94.0	70.5
New NENT+LinkEst	96.5	93.0

Note that the efficiency of the LinkEst process also affects success rates of NENT processes. For example, the success rate of the NENT process using the enhanced NENT and extended LinkEst processes outperforms that using the enhanced NENT and original LinkEst processes by 10%. The reason is that routing paths in those cases were generated statically and in a NENT process new nodes and BS nodes communicate with each other by transmitting control messages. If the LinkEst process is inefficient, routing paths for these control messages will not be created in time. In such a condition, NENT processes of new nodes will be delayed.

As we can see in Fig. 5.5, the required times for nodes to establish links with all of its neighbors using extended LinkEst process can be reduced by 2.57 (i.e., 27.70/10.77 in Table 5.2) times on average. The reason why the enhanced NENT process increases the required time to establish links with all neighboring nodes is that, as mentioned previously, the enhanced NENT process may delay several MSH-NCFG message transmissions to avoid colliding sponsor nodes' MSH-NCFG messages. Thus, since the LinkEst process also uses MSH-NCFG messages to exchange information elements, the required time to establish links with all neighboring nodes will be increased due to delayed MSH-NCFG messages.

We also show average success rates of NENT processes under different scheme combinations in Table 5.1 and average required times for LinkEst processes in Table 5.2. The remaining failure cases result from the fact that a new node's MSH-NENT:NetEntryRequest message is successfully received by its sponsor node but gets collided on other neighboring functional nodes. In such a case, those neighboring functional nodes cannot know the existence of the sponsor node, and thus severe MSH-NCFG collisions cannot be avoided. A new node can deal with this problem by changing its CS node after failing to perform the NENT process with the current CS node for a pre-defined number. Such mechanism is ignored currently in our implementation to reduce design complexity.

Table 5.2: Average times to establish links with all neighboring nodes for a node

Scheme\Case	50-node Case		100-node Case	
	Avg. required time (sec)	Std. dev.	Avg. required time (sec)	Std. dev.
Standard	11.38	1.003	18.94	1.274
New NENT	27.78	2.128	27.70	1.847
New LinkEst	7.11	0.586	7.18	0.471
New NENT+LinkEst	19.14	1.448	10.77	0.734

5.1.4 Summary

In this summary, we identify the reasons that may cause the NENT process to fail in dense IEEE 802.16 mesh networks and propose a scheme to solve this problem. We also show that the LinkEst process defined in the current standard is time-consuming in dense IEEE 802.16 mesh networks and a careful implementation is required to avoid deadlock situations. To address these problem, we propose an extension to the current standard to facilitate the LinkEst process while avoiding deadlock situations, According to our simulation results, on average our enhanced NENT process outperforms the original NENT process by 22.5% in dense networks. As for our extended LinkEst process, on average it speeds up the original LinkEst process by 2.57 times in dense networks.

5.2 The Proposed Two-phase Holdoff Time Scheme

In the IEEE 802.16 mesh CDS-mode network, nodes' holdoff times affect not only the efficiency of MAC-layer scheduling but also the success of network initialization. Network initialization is important because a node must successfully initialize and attach itself to the network before it can start transmitting and receiving data. When designing a new holdoff time scheme, it is important to ensure that network nodes under the new scheme can still successfully attach themselves to the network. When evaluating a holdoff time scheme, three aspects must be carefully considered: the success of network initialization, the efficiency of MAC-layer scheduling, and the fairness of resource sharing.

In this section, we first study how the holdoff times of nodes affect the the network initialization of an IEEE 802.16 mesh CDS-mode network and then propose a two-phase holdoff time scheme to solve this problem. The proposed scheme uses different holdoff time values for the network initialization phase and data scheduling phase. Thus, the success of network initialization can be guaranteed while good scheduling performances can be achieved. In addition, the static and dynamic holdoff time schemes proposed in Chapter 3 can be used in the second phase of this scheme to maximize network performance. In the following, we first explain the effect of the holdoff time value using the simulation results of four fixed-value holdoff time setting schemes. We then explain the necessity of the two-phase holdoff time scheme.

5.2.1 The Effect of the Holdoff Time Value

We use the ten random connected topologies generated in Section 4.2.5 to compare the performances of all studied holdoff time schemes in general networks. The simulation setting used here is the same as that used in Section 4.2.5. In [22], we pointed out two reasons why the initialization of an IEEE 802.16 mesh network may fail. In this section, we applied the revised network initialization process proposed in [22] to all studied schemes, including the original fixed-value schemes. This revised process can significantly alleviate the MSH-NCFG message collision problem.

After this revised process is applied, message collisions now only result from excessive MSH-NCFG message transmissions by a new node's neighboring nodes. A typical example is shown in Fig. 5.6. Suppose that node C is a new node trying to enter the network, and nodes A and B are its neighboring nodes that have attached themselves to the network. The dotted circles represent the signal coverage of nodes A and B, respectively. Before node C attaches itself to this network, nodes A and B transmit their own MSH-NCFG messages without considering whether their MSH-NCFG messages can be successfully received at the location of node C. Consequently, many MSH-NCFG messages transmitted by nodes A and B may be collided at node C. However, since node C so far has not been a functional node in this network, such message collisions does not hinder node C's normal operation at this moment.

However, in case nodes A and B transmit their own MSH-NCFG messages very frequently (for example, these two nodes use a very small holdoff time value to schedule their MSH-NCFG message transmissions), it is very likely that node C cannot successfully receive any MSH-NCFG message transmitted by these two nodes. In this condition, node C cannot proceed its network initialization process because it cannot obtain necessary information required to start its network initialization process.

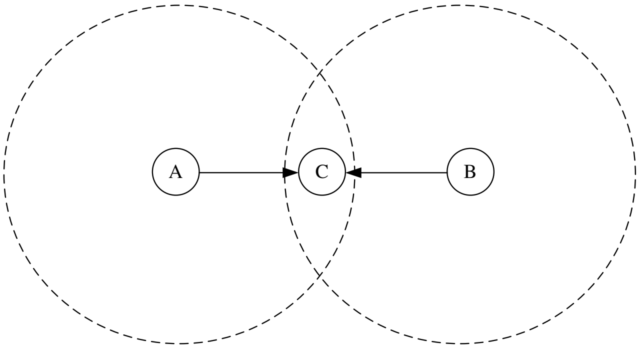


Figure 5.6: Example of MSH-NCFG message collisions

The other reason that causes the network initialization process to fail is the absence of routing paths from a new SS node to a BS node. For an SS node, the success of its network initialization process relies on the availability of a routing path from itself to the BS node. On performing the registration procedure (one of the necessary procedures in a node's network initialization process), a new node must have a routing path to communicate with a BS node. If no available routing path exists, the new node's network initialization process will fail.

To eliminate the effect of the above routing problem from the performance results, we adopted a design to guarantee that 1) every new SS node has a sponsor node and 2) every new SS node has a routing path to the BS node. To provide such a guarantee, we first generated routing paths among all nodes using Dijkstra's shortest path algorithm. Then, we let an SS node choose the first-hop node on its routing path to the BS node as its sponsor node. Such a design guarantees that when an SS node is performing the registration procedure, at least one routing path exist for the SS node to communicate with the BS node. With this design, the problem that network initialization processes fail due to lack of routes to the BS node no longer exists. Thus, the simulation results reflect solely the effect of different holdoff time values rather than the mixed effects of different holdoff time values and the used routing protocol.

In the following, we compare four different fixed-value holdoff time setting schemes. As mentioned previously, the standard regulates that the holdoff time base value be set to 4. Therefore, here we set the holdoff time base value used by all schemes to 4 while varying the holdoff time exponent value used by these schemes from 0 to 3. The resultant holdoff time values are thus 16, 32, 64, and 128, respectively.

Table 5.3 shows the performances of the four fixed-value holdoff time schemes. In

Table 5.3: The performances of the four fixed-value holdoff time setting schemes

	SRNI	ATOUN		ATHPT (ms)		NetUI	
	Avg.	Avg.	Std. dev.	Avg.	Std. dev.	Avg.	Std. dev.
Holdoff Time 16	93%	0.481	0.164	47.423	25.220	95.336	6.083
Holdoff Time 32	98%	0.363	0.165	66.453	15.706	150.262	8.252
Holdoff Time 64	100%	0.237	0.131	104.146	7.518	226.507	10.077
Holdoff Time 128	100%	0.132	0.077	192.719	4.903	318.771	10.629

total, fifty runs of simulations were conducted. The success rate of network initialization (SRNI) is defined in (5.1).

$$SRNI = \frac{NC_{success}}{NC_{total}}, \tag{5.1}$$

where $NC_{success}$ denotes the number of cases in which the network succeeds in initialization and NC_{total} denotes the number of total cases. The success of a network initialization is defined as follows. For a network case, if all of its nodes successfully initialize and attach themselves to the network, the initialization of this network case succeeds. In contrast, if any node fails to perform its initialization and attachment procedures, the initialization of this network case fails. The ATOUN, ATHPT, and NetUI metrics are defined in Section 4.2.1. Each value presented here is the average across the fifty simulation runs.

From Table 5.3, one sees that when the holdoff time value decreases, the ATOUN-Avg. increases and both the ATHPT-Avg. and the NetUI-Avg. decrease. As discussed before, all of these trends are expected and reasonable. These trends show that using a smaller holdoff time value results in better performances when the network is un-congested.

However, the SRNI results reveal a serious problem when small holdoff time values are used. One sees that using large holdoff time values (e.g., 64 and 128) results in a 100% success rate of network initialization. However, using small holdoff time values (e.g., 16 and 32) results in a success rate less than 100%, which means that some SS nodes cannot successfully initialize and attach themselves to the network.

These simulation results show that a fixed holdoff time value cannot provide good scheduling performances while guaranteeing the success of network initialization processes. Based on this observation, we propose a new two-phase holdoff time scheme to achieve both of the above goals. In the following, we explain the proposed two-phase holdoff time scheme in detail.

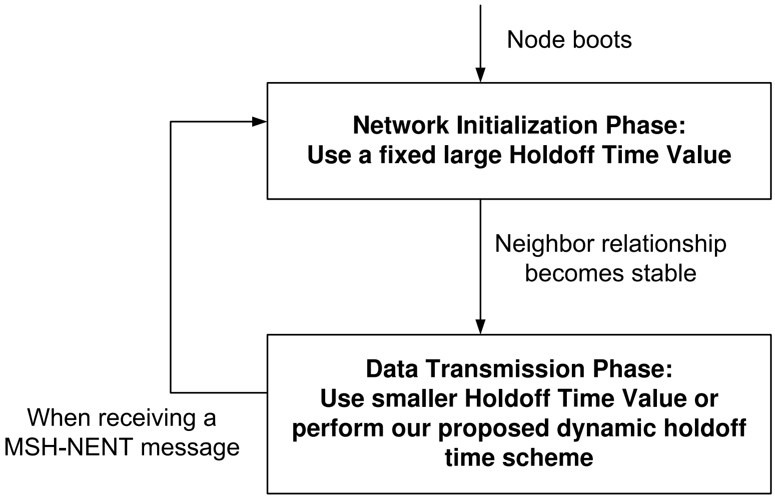

Node boots
Network Initialization Phase:
Use a fixed large Holdoff Time Value
Neighbor relationship becomes stable
Data Transmission Phase:
Use smaller Holdoff Time Value or perform our proposed dynamic holdoff time scheme
When receiving a MSH-NENT message

Figure 5.7: The proposed two-phase holdoff time scheme

5.2.2 Design of the Proposed Two-phase holdoff Time Scheme

In Section 5.2.1, we show that using a fixed holdoff time value cannot guarantee the success of network initialization while providing good scheduling performances. Note that using large NCFG holdoff times for nodes will generate large transmission intervals between consecutive MSH-NCFG message transmissions. Because the control messages of the link establishment process should be transmitted over MSH-NCFG messages, this intuitive scheme will greatly increase the time required for establishing links. However, a link is the entity for scheduling minislot allocations and important to providing QoS support at the MAC-layer. (The details will be explained in Section 5.3). Using large NCFG holdoff times reduce the scheduling flexibility of minislot scheduling and QoS provisioning.

For this reason, in this section we propose a two-phase holdoff time scheme that uses different holdoff time values for the two phases. Fig. 5.7 shows the operation of the proposed scheme. In this scheme, after being powered on, each node initially stays in the *network initialization* phase. It remains in that phase until all of the nodes in its two-hop neighborhood have successfully initialized and attached themselves to the network. When this condition is met, the node then enters the *data transmission* phase.

Because every node should succeed in initializing and attaching itself to the network, when a node is still in its network initialization phase, the proposed scheme sets its holdoff time to a large value (e.g., 64 or higher) to eliminate the potential hidden terminal problem. This design ensures that the network initialization processes of all nodes will eventually succeed. After all nodes have successfully attached themselves to the network, they will have switched their phases to the data transmission phase. In this phase, the hidden terminal problem no longer occurs because the neighbor relationships among all nodes have been known and stabilized. Thus, a small holdoff time value can be used in this phase to improve MAC-layer scheduling performances.

In this scheme, a node uses only its local knowledge to determine when to switch from the network initialization phase to the data transmission phase. According to the standard, each node should maintain a node list to record the scheduling information of the nodes in its two-hop neighborhood. This node list is an input to the pseudo-random election algorithm for scheduling the node's control message transmissions. Assume that for each node the network operator has given it the total number of nodes in its two-hop neighborhood. (This assumption can be easily met for a static network, where dynamic fading effects are not significant.) With this information, each node can locally determine when it can switch to the data transmission phase by comparing the number of nodes currently in its node list with that given by the network operator. When these two numbers match, it can safely enter the data transmission phase for better performances.

Another approach is that, each node can automatically switch itself to the data transmission phase, if it does not receive any MSH-NENT messages for a long period. This means that, most of its neighboring nodes have been joined the network; thus, it can safely switch itself to the data transmission phase. However, as shown in Fig. 5.7, if it receives a MSH-NENT message, it should switch itself back to the network initialization phase to avoid excessive MSH-NCFG message collisions. In addition, based on this design, when a new node cannot listen to any MSH-NCFG messages from neighboring operational nodes, a new node can transmit a MSH-NENT message without specifying the chosen sponsor node ID. This will force neighboring operational nodes to switch themselves back to the network initialization phase and use larger NCFG holdoff times, which can increase the probability that this new receives MSH-NCFG messages to proceed its network entry process.

5.2.3 Summary

In this section, we study the relationship between the holdoff time value and the success rate of network initialization for the IEEE 802.16 mesh CDS-mode network. The simulation results show that using smaller holdoff time values in the network initialization phase will decrease the success rate of initializing an IEEE 802.16 mesh CDS-mode network. To solve this problem, we propose a two-phase adaptive holdoff time scheme that can use larger holdoff time values, when nodes are joining the network, and use smaller holdoff time values, after all nodes have joined the network. The proposed two-phase holdoff time scheme can both guarantee the success of network initialization and achieve better data scheduling performance.

5.3 Enhancement of the Data Scheduling Process in the IEEE 802.16 Mesh CDS Mode

In this section, we point out the drawbacks of the THP design used in the IEEE 802.16(d) mesh CDS mode and propose three schemes to address them. The performance of the three proposed schemes is evaluated using simulations. Our simulation results show that the proposed schemes can effectively improve the data scheduling efficiency of the IEEE 802.16(d) mesh CDS mode and thus improve the network quality experienced by application programs. In addition, the proposed schemes are generic and allow advanced scheduling algorithms to operate on top of them to support better Quality-of-Service (QoS).

5.3.1 The THP of the IEEE 802.16(d) Mesh DS Mode

Procedure Description

The THP of the CDS mode uses the Mesh Distributed Scheduling (MSH-DSCH) messages to exchange the information elements (IEs) required for establishing a data schedule. The detailed steps are explained here: To request a data schedule, the requesting node first transmits a request IE and the corresponding availability IEs to its intended receiving node (referred to as the granting node) using an MSH-DSCH message. A request IE specifies the number of mini-slots demanded by the requesting node, while an availability IE specifies a set of consecutive mini-slots from which the granting node is allowed to choose.

Upon receiving the MSH-DSCH message, the granting node should allocate mini-slots to service the requesting node's request within the mini-slot set specified by the received availability IEs. If the granting node cannot satisfy this request, it can either ignore this request or allocate fewer mini-slots than those specified in the request IE. After finishing the mini-slot allocation process, the granting node sends the requesting node a grant IE describing the allocated mini-slots for the request IE. Finally, on receiving the grant IE, the requesting node broadcasts a confirm IE, which is simply a copy of the grant IE, to acknowledge this mini-slot allocation. During this THP, nodes neighboring to the requesting and granting nodes can overhear the exchanged IEs. Thus, they know when this data schedule will take place and will suspend their data transmissions at that duration to avoid packet collisions.

The Formats of the Three IEs used in THP

The detailed formats of the three IEs used in a THP are presented here. A request IE is represented by a 3-tuple (LinkID, DL, DP), where LinkID denotes the ID of the link over which the requesting node is requesting bandwidth. DL and DP denote the number of consecutive mini-slots per frame demanded by the requesting node and the number of consecutive frames demanded by the requesting node, respectively.

A grant (confirm) IE is represented by a 7-tuple (LinkID, M_Start, M_Range, SFN, D, P, CH), where LinkID has been explained previously. The M_Start field denotes the starting mini-slot number of a mini-slot allocation in a frame and the M_Range field denotes how many consecutive mini-slots in a frame are allocated to a request IE. The SFN field denotes the starting frame number of a mini-slot allocation and the P field indicates in how many consecutive frames this mini-slot allocation will remain valid. The D field specifies whether this IE is a grant IE or a confirm IE, and the CH field specifies the ID of the channel that the granting node uses.

Disadvantages of THP

The THP defined by the mesh DS mode has two disadvantages. First, during a THP, the requesting and granting nodes are only allowed to allocate a "continuous" mini-slot allocation. Recall the grant IE format explained in Section 5.3.1. The 4-tuple (M_Start, M_Range, SFN, P) only can specify a continuous mini-slot allocation. Thus, the granting node is only allowed to allocate a mini-slot allocation for serving the requesting node's bandwidth need. Such a restriction significantly decreases the flexibility of mini-slot scheduling of the mesh DS mode.

Second, in the mesh mode, traffic flows are classified into connections, each of which can have a different QoS requirement. According to [19], the identifier of a connection is composed of a *"LinkID"* field and a set of QoS parameters. The former specifies the ID of the link that is used to service this connection, while the latter describes the QoS requirements of a connection. Outgoing packets of a connection should be transmitted over the mini-slots allocated to the link serving this connection because the THP establishes a data schedule for a link rather than for a connection.

However, one sees that the format of the request IE does not specify the connection and the QoS requirements associated to this bandwidth request. This means that, when a granting node is serving a request IE, it does not know which connections this request IE serves and the QoS needs of these connections. Lacking such information, the granting node cannot employ advanced QoS-aware scheduling algorithms to efficiently utilize link bandwidth and serve traffic flows.

To solve these two problems, here we propose three schemes to improve the data scheduling efficiency of the THP. The first is the multi-grant (MG) scheme, which allows nodes to allocate multiple mini-slot allocations during a THP; the second is the multi-request (MR) scheme, which allows the granting node to schedule link bandwidth using the QoS requirements associated to each request IE; and, the final one is the "multi-request-multi-grant" (MRMG) scheme, which combines the MR and the MG schemes. The details of these schemes are explained in Section 5.3.2.

5.3.2 The Proposed Scheduling Schemes

In this section, we explain the three proposed scheduling schemes and a basic scheduling scheme in details. The basic scheme is used to generate baseline performances, which are compared with those of the proposed schemes. The details of these schemes are presented below.

The Basic Scheme

Using the basic scheme, a network node establishes a link to each of its neighboring nodes. Packets destined to the same node are served by the same link and transmitted in the first-come-first-serve order. Thus, the basic scheme cannot provide QoS guarantee for specific traffic flows. In the basic scheme, a node can negotiate only a single data schedule during a THP.

The MG Scheme

Due to the restriction of continuous mini-slot allocation, the basic scheme cannot efficiently utilize network bandwidth. As shown in Fig. 5.8(a), suppose that node A is requesting node B for a mini-slot allocation that occupies four mini-slots per frame and lasts eight frames (in total 32 mini-slots). In the basic scheme, because node B is only allowed to transmit a data schedule during a THP, the best it can do is to allocate a mini-slot allocation occupying the most mini-slots among all continuous mini-slot blocks that are still available. In this example, node B can only choose a mini-slot allocation that occupies 4 mini-slots per frame and lasts 4 frames for node A. As a result, node A just obtains 16 mini-slots, which is only a half of the mini-slots that it needs. To address this problem, we propose the MG scheme that can satisfy a bandwidth request with multiple separated mini-slot allocations.

In the MG scheme, on receiving a request IE, a granting node repeatedly allocates multiple mini-slot allocations to serve the request IE until one of the following three conditions holds: 1) the bandwidth need of this request IE has been fulfilled; 2) the granting node has no available bandwidth to allocate data schedules; and 3) the number of generated grant IEs has been equal to a pre-specified G_{Thres} value. The G_{Thres} parameter defines the maximum times that a granting node is allowed to perform the mini-slot allocation process for a request IE. The value of G_{Thres} should be properly set based on network load to prevent a network node from monopolizing network bandwidth.

Fig. 5.8(b) shows the advantage of the MG scheme. Consider the same example discussed above. In the MG scheme, node B can collect several smaller fragmented mini-slot blocks to satisfy node A's bandwidth need. In this example, it allocates 4 mini-slot allocations (in total 32 mini-slots) to node A during a THP. The MG scheme outperforms the basic scheme on bandwidth utilization and application throughputs because in this scheme a requesting node can obtain as many mini-slot allocations as possible to meet its needs using only one THP. Besides, the MG scheme also reduces the packet delay time experienced by application programs, as compared with the basic scheme. This is because, using the MG scheme, a node can reduce the number of THPs required to meet its bandwidth needs.

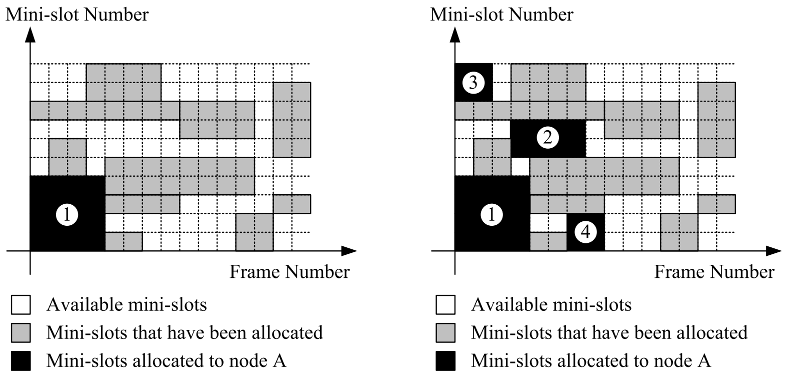
The MR Scheme

Compared with the basic scheme, the MG scheme greatly improves the scheduling efficiency for a bandwidth request. Under these two schemes, however, a requesting node can only use one mini-slot allocation to represent its bandwidth needs during a THP. Such a design results in long scheduling latencies when multiple connections are active, because nodes using this design are only allowed to process a bandwidth request in a THP. As the number of active connections increases, each connection needs to wait a longer time to establish its data schedule. To solve this problem, we propose the MR scheme that is capable of servicing multiple bandwidth requests of active connections during a THP.

The MR scheme exploits a multi-link design to serve connections. Each link is used to service a group of connections. The main idea of the MR scheme is explained below. On initiating a THP, the requesting node can transmit multiple bandwidth requests (represented by request IEs) for different connection groups to the granting node. Thus, the granting node can receive multiple bandwidth requests issued by the requesting node at the same time and then *batch-process* these requests during a THP. Thus, the MR scheme can reduce the number of THPs required to serve the active connections of a requesting node and decrease the time required to establish data schedules.

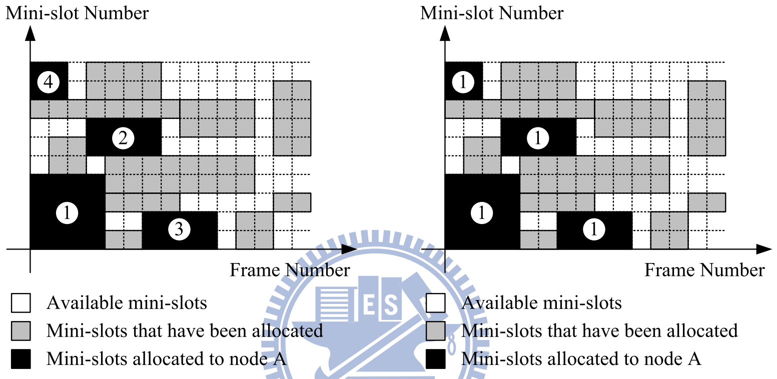
Besides, to solve the second problem of THP discussed in Section 5.3.1, the MR scheme defines a new type of information element (QoS IE) to enhance the QoS support for the THP. A QoS IE is used to notify a granting node of the priority and the QoS requirements of a link, which serves a connection group in the MR scheme. On initiating a THP, in addition to request and availability IEs, a requesting node should transmit to its granting node the QoS IEs associated to links that it requests. Thus, on receiving multiple request IEs at the same time, the granting node can more efficiently serve these bandwidth requests based on the links' priorities and QoS requirements indicated by the received QoS IEs.

The following is an example explaining the advantage of the MR scheme. Assume that node A has established four links to node B. Each link is serving a specific connection group. Each connection group requires 16, 8, 8, and 4 mini-slots, respectively.


Mini-slot Number
1
Frame Number
□ Available mini-slots
▨ Mini-slots that have been allocated
■ Mini-slots allocated to node A
Mini-slot Number
3
2
1
4
Frame Number
□ Available mini-slots
▨ Mini-slots that have been allocated
■ Mini-slots allocated to node A

(a) The basic scheme (b) The MG scheme

Figure 5.8: Examples of mini-slot allocations scheduled by the basic and MG schemes


Mini-slot Number
4
2
1
3
Frame Number
□ Available mini-slots
▨ Mini-slots that have been allocated
■ Mini-slots allocated to node A
Mini-slot Number
1
1
1
1
Frame Number
□ Available mini-slots
▨ Mini-slots that have been allocated
■ Mini-slots allocated to node A

(a) The basic scheme (b) The MR scheme

Figure 5.9: Examples of mini-slot allocations scheduled by the basic and MR schemes

As shown in Fig. 5.9(a), when using the basic or MG scheme, node A needs to launch four THPs to request these mini-slot allocations. However, using the MR scheme node A can simultaneously transmit the request IEs for these connections during a THP. As Fig. 5.9(b) shows, these mini-slot allocations can be scheduled together and transmitted together during a THP. Moreover, since under the MR scheme multiple request IEs of links (and their associated QoS IEs) can simultaneously arrive at the granting node, the granting node has more scheduling flexibility to optimize the use of its link bandwidth.

The MRMG Scheme

The MRMG scheme combines the designs of the MR and MG schemes. Therefore, using the MRMG scheme the requesting node can issue multiple request IEs for different connection groups during a THP. In addition, upon receiving request IEs, the granting

Table 5.4: The parameter setting used in simulations

Parameter Name	**Value**
MSH-CTRL-LEN	8
MSH-DSCH-NUM	8
Scheduling Frames	2
Requested Mini-slot Size	30
Requested Frame Length	32
Modulation/Coding Scheme	64QAM-3/4
Maximum Transmission Range	500 meter
Frame Duration	10 ms
Number of Mini-slots per Frame	220
G_{Thres}	63

node can allocate multiple data schedules for each request IE to optimally satisfy the bandwidth requirements of these request IEs.

5.3.3 Performance Evaluation

In this section, we evaluate the performances of the proposed schemes using the NCTUns network simulator [19]. In the following, we first explain the simulation setting in Section 5.3.3 and then explain the used performance metrics in Section 5.3.3. Finally, we present the simulation results in Section 5.3.3.

Simulation Setting

The main parameters used in our simulations are listed in Table 5.4. We use a 21-node chain network topology as our simulation topology. From the left to the right, nodes in this chain network are named SS(1), SS(2), SS(3),..., and SS(21), respectively. Each node is spaced 450 meters away from its neighboring nodes. The maximum transmission range of each node is set to 500 meters. Nodes $SS(2 + i * 3)$, for $0 \leq i \leq 6$, are selected to generate traffic to their left and right 1-hop neighboring nodes. During simulation, each traffic source node generates four types of traffic: TCP, greedy UDP, UDP with constant-bit-rate (CBR) of 100 KB/sec, and UDP with CBR of 10 KB/sec.

For each evaluated scheme, we run simulations three times, each time using a different random number seed. Each presented performance result is the average of these simulation runs. The total simulated time of each run is set to 200 seconds.

Table 5.5: The performances of the four schemes

	MAC		Application	
	M_O	M_U	AAT (KB/s)	PDT (ms)
MRMG	44.18	57.00%	1504.53	80.160
MR	29.79	38.36%	1235.90	116.034
MG	26.65	34.58%	947.54	181.578
Basic	15.36	19.86%	654.70	382.504

Performance Metrics

Four performance metrics are used to evaluate the proposed schemes. The first is the number of mini-slots obtained by a network node per frame averaged across all nodes (denoted as M_O). M_O represents the scheduling efficiency of a scheme from the perspective of a network node. A larger value of M_O indicates that a node can obtain more mini-slots during a THP.

The second metric is the sum of the mini-slot utilization of all nodes in a node's two-hop neighborhood, averaged across all nodes in a network (denoted as M_U). The mini-slot utilization of a node is defined as the ratio of the number of mini-slots used by the node to the number of total mini-slots available during the interval that its traffic flows are active. In the mesh DS mode, the two-hop neighborhood of a network node is defined as the set composed of the node's one-hop and two-hop neighboring nodes and the node itself. A network node should coordinate its packet transmissions among its two-hop neighborhood to avoid the *hidden-terminal* problem. A high value of M_U represents that network bandwidth can be utilized efficiently, while a low value of M_U represents that network bandwidth is under-utilized due to too conservative scheduling.

In addition to MAC-layer performances, the Average Application Throughput (denoted as AAT) and Packet Delay Time (denoted as PDT) are also used to evaluate the network quality experienced by application programs. The former is defined as the average of the aggregate throughputs obtained by all traffic flows, while the latter is defined as the average of the times required for packets to travel from their source nodes to their destination nodes.

Simulation Results

As shown in Table 5.5, the three proposed schemes outperform the basic scheme on MAC-layer performances. The MRMG scheme on average can obtain 2.88 times mini-slots

Table 5.6: The average bandwidth satisfaction index results

	RT CBR-25KB		RT CBR-3KB		TCP		UDP	
	AAT (KB/s)	ABSI (%)	AAT (KB/s)	ABSI (%)	AAT (KB/s)	ABSI (%)	AAT (KB/s)	ABSI (%)
MRMG	20.20	80.80	2.60	86.67	97.52	97.52	396.28	99.07
MR	23.34	93.36	2.99	99.67	95.79	95.79	373.25	93.31
MG	9.34	37.36	1.69	56.33	79.38	79.38	225.84	56.46
Basic	6.63	26.52	1.77	59.00	69.75	69.75	142.09	35.52

for served flows in a THP, as compared with the basic scheme. One notices that the MR scheme on average can outperform the MG scheme. The reason is explained below. Using the MR scheme the MAC layer is allowed to allocate only one data schedule for a request IE; however, it can issue up to four request IEs (each for a specific connection group) during a THP. Thus, the bandwidth needs of the four connection groups can be processed and satisfied using only one THP. In contrast, although using the MG scheme the MAC layer can allocate more than one data schedules for a request IE, only a request IE can be transmitted and served during a THP. Thus, using the MG scheme the MAC layer should perform four THPs to service the four connection groups, resulting in inefficient use of link bandwidth and larger data scheduling latencies for the four connection groups.

One sees that the MRMG and MR schemes can greatly reduce the packet delay times experienced by traffic flows, as compared with the MG and basic schemes. These results show that the multi-request design can more efficiently serve traffic flows belonging to different connection groups during a THP. Also, the MRMG scheme on average can result in shorter packet delay time than the MR scheme because the former exploits the multi-grant design to satisfy each request IE's bandwidth need as much as possible. Thus, nodes using the MRMG scheme on average can reduce the queuing delay of a data packet because a data packet can be transmitted sooner after being inserted into the link transmission queue.

The simulation results show that the MRMG scheme on average can increase application throughputs by a factor of 2.298 and reduce the packet delay time by a factor of 4.772, as compared with the basic scheme. The reason is that the multi-grant design can maximally satisfy the bandwidth need of a request IE while the multi-request design can efficiently satisfy the bandwidth needs of multiple request IEs simultaneously. Thus, the MRMG scheme, which exploits these two designs, in general can achieve the best performances among all the evaluated schemes.

Regarding the capability of QoS support, we define the average bandwidth satisfaction index (ABSI) to evaluate how well the bandwidth need of each flow is satisfied under a scheduling scheme. The ABSI of a scheduling scheme is computed as follows. We first compute the bandwidth satisfaction index (BSI) per second for each flow using the following equation:

$$\mathrm{BSI}(i) = \frac{\text{the amount of the granted BW in } i\text{'th sec}}{\text{the amount of the requested BW in } i\text{'th sec}}. \tag{5.2}$$

We then compute the average BSI value of each flow (denoted as BSIF) as follows:

$$\mathrm{BSIF}_j = \frac{\sum_{i=1}^{n} \mathrm{BSI}(i)}{n}, \tag{5.3}$$

where n is the number of seconds of flow j's active duration. The ABSI of a simulation run is calculated as the average of all flows' BSIF values. The ABSI result of a scheme is the average of the ABSI results of all simulation runs. We conducted a series of simulations using the previous settings except the traffic pattern. In these simulations, each traffic source node generates four types of traffic: Real-time UDP with CBR of 25 KB/sec, Real-time UDP with CBR of 3 KB/sec, TCP, and greedy UDP, respectively.

Table 5.6 shows the ABSI results of the evaluated schemes. As one sees, the MRMG and MR schemes can serve traffic flows much better than non-MR schemes (such as the MG and the basic schemes). Due to the use of the multi-request design, the MRMG and MR schemes can serve multiple connections well based on their priorities and QoS requirements in a THP. Thus, under these two schemes each connection can obtain a bandwidth more close to what it requires. It is worthy to note that the ABSI values of real-time traffic under the MRMG scheme are a bit less than those under the MR scheme. These results show that the MRMG scheme sometimes exhausts link bandwidth to serve greedy traffic (TCP and greedy UDP). Thus, upon receiving the bandwidth requests from real-time CBR connections, granting nodes using the MRMG scheme may have less link bandwidth to service these requests than those using the MR scheme. One solution is to employ a collaborative bandwidth sharing algorithm on top of the MRMG scheme to guarantee proportional sharing of link bandwidth among different connections. Such an extension, however, is out of the scope of this dissertation and is left as future work.

5.3.4 Summary

In this section, we propose three schemes to improve the data scheduling performances of the IEEE 802.16(d) mesh DS mode. The proposed MR, MG, and MRMG schemes are presented in detail in this section. Our simulation results show that the data scheduling efficiency of the IEEE 802.16(d) mesh DS mode and the network quality experienced by application programs are significantly improved under these schemes.

5.4 Effects of Collaborative Routing Protocols on WMNs

An IEEE 802.16 mesh CDS-mode WMN falls into the infrastructure/backbone WMN category [23] and usually operates with two radios. The nodes in this network can be divided into types: the mesh Access Point (mesh AP) and the mesh client. The mesh AP node acts as a wireless router/switch and is usually composed of two radios. One is the IEEE 802.16 mesh CDS-mode radio, which is used to form the backbone relay network; the other is commodity radios (e.g., IEEE 802.11(a/b/g) radios), which is used to communicate with mesh clients. The mesh client node represents wireless devices equipped with commodity radios that can access the WMN, e.g., laptops, smart phones, and vehicle On-board Units equipped with IEEE 802.11(a/b/g) radios. Such a network architecture is also called a dual-radio-dual-mode WMN [24, 25, 26].

To reduce deployment cost, only a few mesh APs are deployed with wired links to connect to the Internet. Such mesh APs are called Internet gateway nodes. When accessing the Internet, a mesh client in this network first transmits its data to its associated mesh AP using the commodity radio. Then, the associated mesh AP *"wirelessly"* forwards the received data to an Internet gateway node. Finally, upon receiving these data, the Internet gateway node forwards them to the Internet via its wired link.

One key problem to the infrastructure/backbone WMN is that how mesh APs efficiently route upstream data packets (from clients) to an Internet gateway node and downstream packets (from the Internet) to their destination mesh clients. The backbone relay network in an infrastructure WMN can be viewed as a Mobile Ad-hoc Network (MANET). Since it shares several features with MANET, the routing protocols developed for MANET [27] can be applied to the backbone relay network of an infrastructure WMN. For example, Microsoft mesh networks [28] and RoofNet of MIT [29] route packets based on the Dynamic Source Routing (DSR) protocol [30, 31], and several commercial WMN products adopt the Ad hoc On-demand Distance Vector Routing (AODV) protocol [32] as their underlying routing protocols.

Although several sophisticated routing protocols developed for MANET can be used in WMNs, it is unclear whether they would perform better than the routing protocols developed for fixed Internet. WMNs are not exactly the same as MANET. In an infrastructure WMN, APs are fixed and only clients may move. When a client moves and changes its associated AP, the client location database of the WMN can be updated with the new AP. With this database, when a client intends to transmit packets to another client in the same WMN, it first looks up the database to find the current AP with which the destination client is associated and then transmits packets to that AP. After receiving these packets, that AP can forward them to the destination client.

In this condition, the routing paths among APs need not be affected by client movements. Thus, the routing protocols designed for fixed networks may be already good enough for the AP network of an infrastructure WMN. MANET routing protocols generally assume that all network nodes are mobile. For this reason, most of them aggressively broadcast control packets to quickly detect link breakage caused by node movements. However, in an infrastructure WMN, where APs are fixed, the bandwidth consumed by these control packets will be wasted. For these concerns, some WMN products adopt the routing protocols designed for fixed networks. For example, Tropos Networks Inc. uses predictive wireless routing protocol (PWRP), which is similar to OSPF [33], to support a WMN [34].

In the literature, the performances of routing protocols operating in dual-radio-dual-mode WMNs are rarely presented. In the chapter, we use simulations to implement and evaluate the performance of three routing protocols that have been adopted by some commercial WMNs: OSPF, AODV, and STP (Spanning Tree Protocol) [35]. By evaluating and comparing their performances under various network conditions, we reveal the advantages and disadvantages of these routing protocols when they operate in infrastructure WMNs. In addition, to mitigate performance bottleneck on a few Internet gateway nodes in infrastructure WMNs, we implemented a multi-gateway WMN and studied the effectiveness of using multiple gateways on the system total throughput.

5.4.1 The Studied Routing Protocols

The routing protocols evaluated in this chapter can be categorized into three classes: 1) ad hoc routing protocols; 2) spanning tree protocol; and 3) IP routing protocols.

1. Ad-hoc Routing Protocols

Ad-hoc routing protocols are either proactive (table-driven) or reactive (on-demand). Proactive protocols (e.g., DSDV [36]) generate route control packets periodically between nodes. Every node needs to maintain a routing entry for every other node. Each time a periodic route control packet is received, a node recomputes the route derived from the control packet and updates its routing table if needed. The drawbacks of proactive routing protocols are discussed here. First, in the low mobility environment, routes among nodes infrequently change over time; thus, periodically broadcasting control packets will waste link bandwidth. Second, using proactive routing protocol nodes may maintain many routes that will never be used. Control messages exchanged for maintaining these unused routes may greatly waste link bandwidth.

Reactive protocols (e.g., DSR [30, 31] and AODV [32]) trigger the routing path construction only when necessary. To transmit a packet, a node consults its routing table to find a valid route to the destination of the packet. If a valid route can be found, it transmits the packet out. Otherwise, it initiates a route request process for the destination node. When receiving a response for the destination node, the source node generates a valid routing entry for this destination node. The validity of the route is determined by the lifetime specified for it. If the route has not been used for an interval longer than its specified lifetime, it will be considered to be no longer needed and removed from the routing table. Contrary to proactive protocols, reactive protocols maintain routing information only when needing to transmit packets. It reduces unnecessary bandwidth overheads at the cost of spending more time on finding a route.

The Ad-Hoc on Demand Distance Vector (AODV) [32] routing protocol is a representative of the reactive protocols for ad-hoc networks. In AODV, the route discovery and maintenance are triggered by the demands of transmitting packets. The source node initiates the route discovery process by broadcasting the route request (RREQ) only when it tries to send a packet and there is no active route found in its routing table. Except the destination node, each node receiving the RREQ will re-flood it, until the RREQ reaches the destination node. Upon receiving the RREQ, the destination node sends back a route reply (RREP) to the source node through the reverse path of RREQ. In AODV, an alternative way is used to decrease the response time of the route discovery process. If an intermediate node already has the routing information of the destination node, when receiving an RREQ for inquiring a route towards it, this intermediate node can send back an RREP to the source node without further flooding the RREQ.

Currently, AODV is popular and has been adopted by some commercial WMN products. For this reason, we select it as one of the three routing protocols studied in this chapter.

2. Spanning Tree Protocol

The spanning tree protocol (STP) [35] is a loop-prevention method on LANs where multiple bridges/switches may be inter-connected and physical loops may be formed. The leader-election algorithm in STP selects a bridge on a LAN as the root bridge of the spanning tree. Each bridge running STP exchanges its local information using the Bridge Protocol Data Unit (BPDU) message. When the priorities of all bridges combined with their MAC addresses are exchanged over the whole network, the bridge with the highest ID is selected as the root bridge. All ports on the root bridge are known as designated ports. On a link segment, only one attached port can be designated and all others must be blocked. All designated ports are in the forwarding state. A port in the forwarding state is allowed to send and receive data. All of the other bridges are called non-root bridges. A non-root bridge should choose one of its port to be “root port” to send and receive data to/from the root bridge.

Using this method, redundant ports (links) are closed down and packets will not be endlessly spawned and trapped in loops. A closed port can be opened again if the network topology changes and the closed port is needed to form the new spanning tree.

A traditional fixed switch uses one of its ports to connect to another switch via a cable. In a WMN, however, a mesh AP uses its single ad-hoc mode interface to exchange packets with multiple neighboring APs. Therefore, a port in a WMN should be re-defined to be the ad-hoc wireless connection that is used to exchange packets between two mesh APs.

Because the fixed AP network of a WMN functions like a layer-2 LAN with redundant (wireless) links, STP can be applied to these fixed APs to construct a spanning tree (packet forwarding paths) among them without any loop. Since STP can perform self-routing, self-organization, and self-healing, it is fault-tolerant and can cope with node mobility. STP has been widely implemented on switches. Many commercial WLAN AP products have included it as a standard feature so that these APs can readily form an IEEE 802.11 WDS (Wireless Distributed System) [37] when they are set to the bridging mode.

3. IP Routing Protocols

In the Internet, routes are determined by IP routing protocols. Currently, the two most popular routing protocols controlling the routes of Internet traffic are OSPF [33] and RIP [38], which are explained below.

The Route Information Protocol (RIP) is based on the distance-vector algorithm. Routers periodically broadcast their own routing tables to neighboring routers and calculate a shortest path based on the exchanged information to route packets. RIP is not suitable for large networks because it generates many control messages and thus wastes much link bandwidth.

The Open Shortest Path First (OSPF) is a routing protocol developed for IP networks by IETF. In the mid-1980s, because RIP was increasingly incapable of serving large and heterogeneous networks, OSPF was created to replace RIP. OSPF is a link-state routing protocol that relies on flooding of link-state advertisements (LSA) to all other routers within the same hierarchical area. As an OSPF router accumulates link-state information, it uses the Shortest Path First (SPF) algorithm to calculate the shortest paths to all other routers.

Due to the great success of OSPF on the Internet and the fact that the AP network of a WMN is a fixed network, it is natural for people to think that OSPF may also be able to provide good performances in WMNs. For this reason, some commercial WMN products use OSPF as the routing protocol running among mesh APs.

5.4.2 Routing Protocol Design and Implementation

We evaluate and compare the performances of these routing protocols using simulations. The three evaluated protocols are implemented on the NCTUns network simulator

[19], which is a high-fidelity network simulator capable of simulating various protocols used in both wired and wireless IP networks. NCTUns provides a module-based platform for developers to develop new network protocol modules. New protocol modules developed by users can be easily integrated into NCTUns to form new types of nodes. The detailed implementation of the three evaluated routing protocols over NCTUns are presented below.

Two Types of Traffic in a WMN

Each mesh client is associated with a mesh AP via a commodity radio (e.g., the IEEE 802.11(b) infrastructure mode radio). As a mesh client moves, it may disassociate with the current mesh AP and re-associate with a new mesh AP. In our implementation, each mesh AP employs a Wireless Mesh Routing (WMR) module to know which mesh clients are associated with it. WMR records the IP and MAC addresses of mesh clients that are currently associated to the mesh AP on which it installs. The traffic in a WMN can be divided into two types: 1) Internet and 2) Intra-WMN, which is explained below.

A traffic flow is categorized in the Internet traffic if its packets should pass through the Internet gateway node of the WMN, e.g., a TCP connection on a mesh client connecting to a web server in the Internet to download a web page. In this usage, the mesh client is given a gateway IP address so that its packets can be first sent to the mesh Internet gateway, which then further forwards the packets to the Internet. Fig.5.10 shows the protocol stacks of the mesh client, mesh AP, and mesh Internet gateways involved in this type of usage. A traffic flow is categorized in the Intra-WMN traffic if its packets only flow in the WMN, e.g., a VoIP phone call between two mesh clients. In this usage, the source mesh client is given the IP address of the destination mesh client. Fig.5.11 shows the protocol stacks of the mesh client and mesh AP involved in this type of usage.

5.4.3 Address Resolution Protocol (ARP)

Because the AP network of a WMN usually forms the same subnet, to forward packets to a mesh Internet gateway, a source mesh client needs to know the MAC address of the chosen mesh Internet gateway. This can be accomplished by using the Address Resolution Protocol (ARP), which is designed to know the mapping between an IP address and its associated MAC address. Using ARP, the source mesh client first broadcasts an

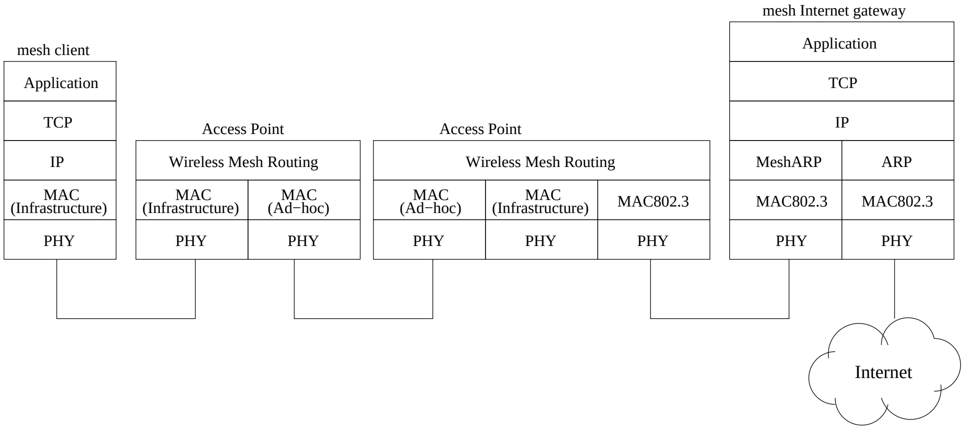


Figure 5.10: The protocol stacks of the mesh client, mesh AP, and mesh Internet gateways (WMN-to-Internet)

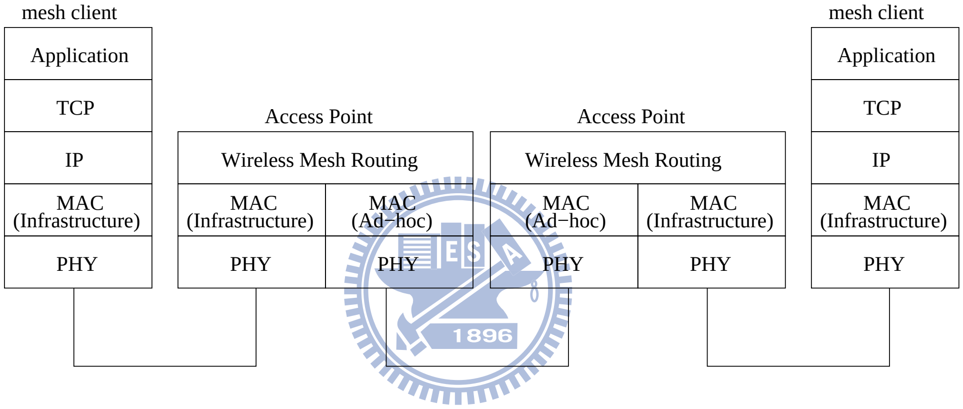


Figure 5.11: The protocol stacks of the mesh client and mesh AP (WMN-to-WMN)

ARP request message, which contains the queried IP address, via its infrastructure mode interface. This message will be received by the mesh AP with which the source mesh client is associated. When a mesh AP receives such a message, it first checks whether the specified IP address is used by its associated mesh clients.

If this is the case, it broadcasts this ARP request message to all of its associated mesh clients via its infrastructure mode interface. When the destination mesh client receives this ARP request message, it sends back an ARP reply message with its own MAC address via its infrastructure mode interface. Upon receiving the ARP reply message, the mesh AP then forwards it to the source mesh client via its ad-hoc mode interface. On the other hand, if the mesh AP finds that none of its associated mesh client uses the specified IP address, it broadcasts the ARP request via its ad-hoc mode interface to other neighboring mesh APs to continue the address-resolution process. If a mesh AP connects to a mesh

Internet gateway, it also sends the ARP request message to the gateway. With this design, a mesh client can obtain the MAC address used by either the mesh Internet gateway or another mesh client.

5.4.4 Routing Procedures

In the following, we present the detailed routing procedures used in the three evaluated routing protocols. The WMR module of a mesh AP is responsible for running the AODV, STP, or OSPF routing protocol. To save space, we only present the routing procedures involved in the “From WMN to WMN” traffic type.

1. AODV

When a mesh AP receives a RREQ (route request) packet, its WMR first checks whether the destination mesh client is associated with the AP. If this is the case, the mesh AP sends the RREQ to the destination mesh client via its infrastructure mode interface. Upon receiving the RREQ, the destination mesh client sends back a RREP to the associated mesh AP, which then sends back the RREP toward the source mesh client via its ad-hoc mode interface. When the RREP is forwarded toward the source mesh client, the WMR modules of intermediate mesh APs will create a valid and active routing entry for the destination mesh client in their routing tables. Such a routing entry includes the MAC address of the destination mesh client and the MAC address of the ad-hoc mode interface of the next-hop mesh AP.

When the RREP comes back to the source mesh client, the source mesh client can start sending its packets. First, the destination MAC address of the packet is filled in with the MAC address of the destination mesh client. Then the packet is sent out via the mesh client’s infrastructure mode interface. The packet will be received by the associated mesh AP via its infrastructure mode interface. The WMR module in the mesh AP checks whether the destination MAC address of the received packet matches the MAC address of any of its associated mesh clients. If this is the case, the packet is sent out through its infrastructure mode interface to reach the specified destination mesh client. Otherwise, the WMR module checks its routing table to find the MAC address of the next-hop mesh AP for this packet.

Before forwarding the packet via its ad-hoc mode interface, the mesh AP encapsulates the packet with a tunneling header. It then fills in the next-hop MAC address with the destination MAC address of this tunneling header. The mesh AP then sends out the encapsulated packet via its ad-hoc mode interface, which will be received by other neighboring mesh APs via their ad-hoc mode interfaces. Based on the destination MAC address in the tunneling header, each of these mesh APs determines whether it should 1) drop the packet, 2) send the packet to the destination mesh client should it is associated with the AP, or 3) check its routing table to continue forwarding the packet to another mesh AP. In the second case, the mesh AP first strips off the tunneling header and then transmits the packet to the destination mesh client. In the third case, the mesh AP replaces the tunneling header with a new one before forwarding the packet. The above steps are repeated until the packet reaches the destination mesh client.

2. Spanning Tree Protocol (STP)

The WMR in each mesh AP runs the spanning tree protocol (STP) to form a spanning tree among them. As described previously, because in a WMN a mesh AP can connect to multiple neighboring mesh APs, the connection between two neighboring APs is considered as a switch port and associated with a port state. The WMR does not know how many ports it has in advance. Instead, when receiving a STP control packet from a neighboring AP, it checks whether this port (i.e., connection) has been created before. If not, it dynamically adds this port into the port list. This port-closing operation repeats until a loop-free connection tree is built among mesh APs. The resulting loop-free connection tree can be used to route packets among mesh APs without generating the route loop problem.

Using STP, a mesh client transmits packets to another mesh client using its infrastructure mode interface. Thus, these packets first arrive at the mesh AP with which this mesh client is associated. Upon receiving a data packet from its associated mesh client, a mesh AP first encapsulates it with a tunneling header and transmits it out via an open port on its ad-hoc mode interface. STP uses the learning-bridge protocol [39] to determine which ports should be open and closed.

Initially, every mesh AP should run the learning-bridge protocol when it is booted. When a mesh AP does not know which port can be used to forward a packet, it sends a copy of the packet to each of its ports except the one from which this packet is received.

The learning-bridge protocol learns and installs routing entries when forwarding packets. Thus, when a mesh AP receives a packet from a neighboring AP, it records the neighboring mesh AP as the next-hop mesh AP for the mesh client specified in the source MAC address of the packet. Then, the mesh AP removes the tunneling header from the packet. If it finds that the destination MAC address of the packet matches the MAC address of one of its associated clients, it should transmit this packet via its infrastructure mode interface to the destination mesh client.

When the packet is eventually received by the destination mesh client, the destination mesh client may send a reply packet back to the source mesh client via its infrastructure mode interface. Since the intermediate mesh APs have learned and installed the routing entries for the source mesh client, this reply packet is forwarded using unicast rather than flooding. When the reply packet is on its way back to the source mesh client, these intermediate mesh APs learn and install routing entries for the destination mesh client. This way, after a packet exchange between the source and destination mesh clients, the unicast routing path between them on the spanning tree is established and the initial blind-flooding is no longer needed.

3. OSPF

The WMR in each mesh AP runs the OSPF protocol to build shortest routing paths to all other mesh APs. Each mesh AP periodically broadcasts hello packets to its neighboring mesh APs so that network topology changes can be detected. If there is a topology change, an involved mesh AP will flood its updated Link State Advertisements (LSAs) to the whole network and all other mesh APs will use the latest LSA to update their shortest path trees and routing tables. A mesh AP has two kinds of neighbors: mesh APs and mesh clients. A neighbor addition/removal of any kind will trigger the mesh AP to flood an updated LSA to the network. A mesh client does not broadcast hello packets and it is viewed as a neighbor by only the mesh AP with which it is associated. If a mesh AP receives a LSA from a neighboring mesh AP and finds that this LSA contains a mesh client that is associated with it, it will assume that the mesh client has moved out of its own radio coverage and into the radio coverage of the new mesh AP. Thus, it should remove the mesh client from its own neighbor list and flood a LSA to the network to announce this topology change.

When a mesh AP floods a LSA, the MAC addresses of the mesh clients associated with it are included in the LSA. Using this design, the MAC address of every mesh client in the WMN is propagated throughout the network and known by every mesh AP. Since each mesh AP has a routing entry for every mesh client, a mesh client's packets can be forwarded toward the mesh AP with which the destination mesh client is associated. To preserve the original source and destination MAC addresses, as in the AODV and STP approaches, intermediate mesh APs use a tunneling header to specify the next-hop mesh AP for a packet.

5.4.5 Multi-Gateway Wireless Mesh Networks

A WMN is commonly used as an Internet access network by which mesh clients send/receive information to/from the Internet. If a WMN has only one mesh AP to connect with the Internet gateway node, all traffic flows will need to be merged at that mesh AP, which apparently limits the aggregate throughput of the WMN (from and to the Internet) to only the bandwidth that can be achieved by an IEEE 802.11(b) interface.

To address this problem, we designed and implemented a multi-gateway WMN, where multiple mesh APs can connect to a mesh Internet gateway to share Internet traffic load among them. (These mesh APs are called the "gateway mesh AP" below.) Each of these gateway mesh APs uses a high-speed (e.g., 100 Mbps or Gbps) link to connect to the mesh Internet gateway to ensure that the used link is not a performance bottleneck of the WMN. In our design, as Fig. 5.12 shows, when mesh clients broadcast their ARP requests to ask the MAC address of the gateway (they all use the same gateway IP address, which is 1.0.1.1 in this example), different mesh clients may get different gateway MAC addresses. By using this design, Internet traffic generated by mesh clients can be distributed to different gateway mesh APs without overloading a single gateway mesh AP. This will make a WMN more scalable with the number of mesh clients.

The internal design is presented here. After a mesh client broadcasts an ARP request, mesh APs that receive this ARP request will flood it toward the mesh Internet gateway. Since an ARP request is a broadcast packet, it may be cloned by mesh APs and its multiple copies may reach the mesh Internet gateway. To avoid wasting bandwidth, mesh APs use the sequence number carried in the ARP request to detect and avoid sending redundant packets. However, with this design each of the interfaces of the mesh Internet

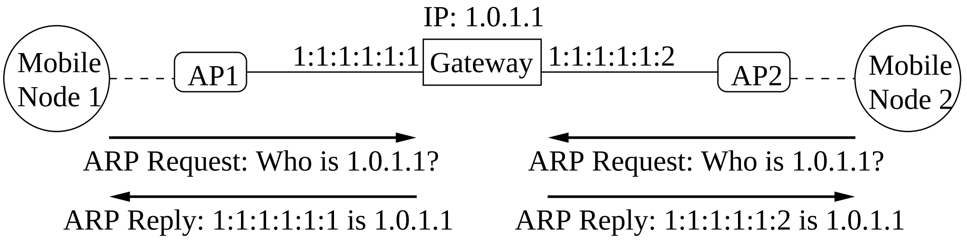

IP: 1.0.1.1
Mobile Node 1
AP1
1:1:1:1:1:1
Gateway
1:1:1:1:1:2
AP2
Mobile Node 2
ARP Request: Who is 1.0.1.1?
ARP Reply: 1:1:1:1:1:1 is 1.0.1.1
ARP Request: Who is 1.0.1.1?
ARP Reply: 1:1:1:1:1:2 is 1.0.1.1

Figure 5.12: The ARP request and reply procedure in a multi-gateway WMN

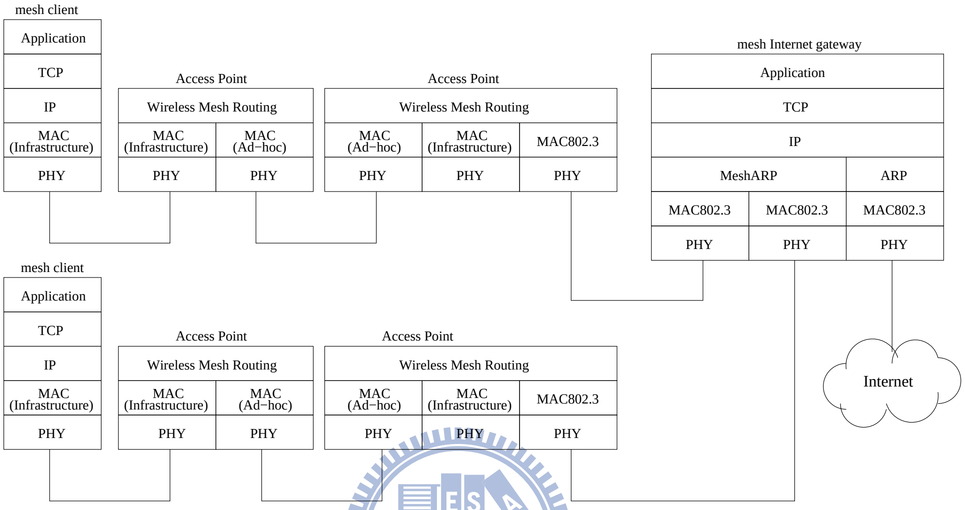


Figure 5.13: The internal protocol stack of a multi-gateway WMN

gateway may still receive one copy of the same ARP request. (Each of these copies is transmitted along a distinct path.)

When receiving an ARP request from an interface, the gateway performs two operations: 1) First, it puts the MAC address of this interface in the ARP reply and uses the same interface to send back the ARP reply; 2) Second, it creates a (mesh client source MAC address, interface ID) entry in its routing table. Fig. 5.13 shows the protocol stacks of gateway mesh APs and the mesh Internet gateway. The MeshARP module is responsible for performing the two operations described above.

Executing the first operation enables a mesh client to use its nearby gateway mesh APs to relay its traffic to the Internet. This allows the load of outgoing Internet traffic to be balanced among different gateway mesh APs. As discussed previously, the same ARP request may reach the gateway more than once (one request from each of its interfaces) and this will cause the gateway to send back a reply for each of them. However, this behavior does not generate problems. When the source mesh client receives multiple ARP replies (each carrying a different gateway MAC address), it can simply use the MAC address carried in the first reply as the MAC address of the gateway. Normally, such a reply carries the MAC address of the gateway interface with a lighter load, which is exactly the interface that should be used for load-balancing purposes. On the other hand, executing the second operation allows incoming Internet traffic to be routed to different mesh clients via different gateway mesh APs. This allows the load of incoming Internet traffic to be balanced among different gateway mesh APs.

5.4.6 OSPF with the Expected Transmission Count Metric

The Expected Transmission Count metric (called ETX) is designed for RoofNet [29] at MIT to enhance the performance of the ad-hoc routing protocol. The ETX metric is used for a routing protocol to choose a routing path with a higher end-to-end throughput. In the RoofNet, DSR is modified to work with the ETX metric and it is shown that using the ETX metric can improve the end-to-end throughput between a pair of nodes. To see how ETX can improve the OSPF routing protocol in WMNs, we implemented a version of OSPF with ETX support.

To support the ETX metric, each mesh AP records the number of hello packets it receives during a period and uses it to calculate the delivery ratio of each neighbor. The mesh AP stores these delivery ratios in its LSA packets to inform other mesh APs of the delivery ratios of its neighbors. These delivery ratios are used as link weights when a mesh AP computes the shortest paths to all other mesh APs. By this method, a smallest ETX path tree can be constructed, which can be used to find a high-throughput routing path between a pair of mesh clients.

5.4.7 Performance Evaluation

In this section, we evaluate the performance of WMNs using simulations. The simulations are performed on the NCTUns network simulator. Each case is simulated 20 times with different random mesh client locations. The average of these simulation results is presented in this section. The total simulation time of each case is 200 seconds. We only recorded the TCP results during the last 100 seconds to avoid the influence of the TCP startup behavior on the logged end-to-end throughputs. As Fig. 5.14 shows, 25 mesh APs are deployed in a 5x5 grid network. The distance between two neighboring mesh APs is

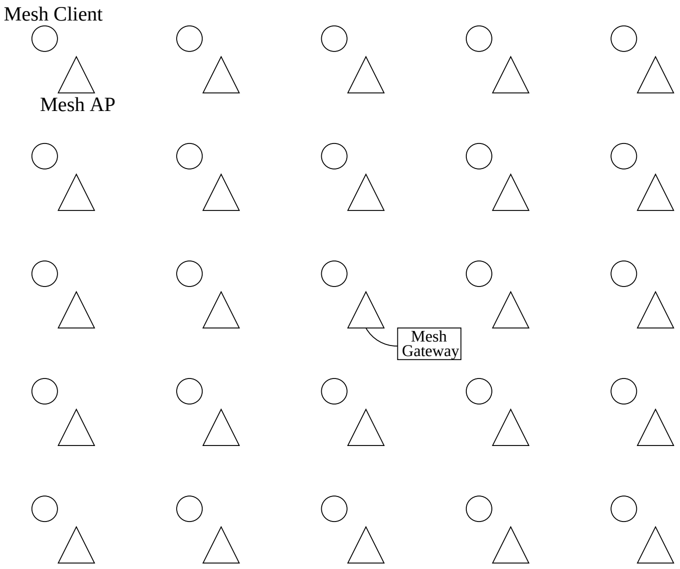


Figure 5.14: The simulation topology

200 meters.

Each mesh AP has two IEEE 802.11(b) interfaces. One operates in the ad-hoc mode for forwarding packets among mesh APs and the other operates in the infrastructure mode for serving mesh clients. The transmission range and the interference range of these wireless interfaces are set to be 250/550 meters, respectively, which are used in the ns-2 network simulator [40] and commonly used in the literature. The ad-hoc mode and infrastructure mode interfaces use different channels to avoid interferences. The three routing protocols (i.e., OSPF, STP, and AODV) are run on mesh APs for performance evaluation. The mesh AP at the center of the field connects to the mesh Internet gateway via a link with the 100 Mbps bandwidth. There are 25 mesh clients scattered at random locations in the field. Each mesh client has an infrastructure mode IEEE 802.11(b) interface and uses it to send/receive packets to/from its associated mesh AP. For readers' convenience, in Fig. 5.14 a mesh client is drawn at a location close to a mesh AP. However, in simulations since mesh clients are randomly placed in the field, the client-AP association relationship is not always one-to-one as shown in the figure. Multiple mesh clients can be associated with a mesh AP and a mesh AP does not have any mesh client associated with it.

Table 5.7: The total download throughput in the one-to-multi downlink traffic case

Protocol	OSPF	STP	AODV
Total Download Throughput (KB/s)	253.8	255.7	245.6
Standard Deviation	15.3	14.2	16.4

Table 5.8: The number of established and stable connections in the one-to-multi downlink traffic case

Protocol	OSPF	STP	AODV
Number of Established Connections	24.8	24.7	24.88
Standard Deviation	2.5	2.4	2.6
Number of Stable Connections	24.65	24.4	17.06
Standard Deviation	2.4	2.3	2

One-to-Multi Downlink Traffic Configuration

TCP is widely used in current Internet applications (such as, FTP, HTTP, email, etc.) to reliably transmit data across heterogeneous networks. Internet users usually download data from the Internet through an Internet gateway. Thus, we study a downlink TCP traffic case here. There is a TCP receiver (rtcp) running on each mesh client. Twenty five TCP senders (stcp) run on the mesh Internet gateway and each one greedily sends traffic to the TCP receiver running on each mesh client. That is, the 25 greedy TCP traffic flows compete for the bandwidth of the WMN. All mesh clients are fixed in this simulation case.

Tab. 5.7 shows the total download throughput of the WMN using the three evaluated protocols. one can see that the three routing protocols provide approximately the same total download throughput; however, as shown in Tab. 5.8, AODV results in the least number of stable connections in the WMN. (In this section, a TCP connection is defined as an unstable connection if the receiver program of this TCP connection does not receive data for $\frac{T_{sim}}{2}$ seconds, T_{sim} is the total simulation time.)

An unstable TCP connection is due to excessive triggering of the TCP congestion control on the sending side of this connection, which prevents the TCP sender from sending out data for a long period of time. Due to the design of TCP congestion control, a packet loss will trigger TCP congestion control and many packet losses may result in a long transmission timeout.

To explain why AODV achieve the least number of stable TCP connections in a WMN, we studied its protocol design and the effect of its parameters. We found that in a

Table 5.9: Number of established and stable connections with AODV

ACTIVE ROUTE TIMEOUT	3	10	30	50	100
Number of Established Connections	24.88	24.85	24.5	23.95	23.2
Standard Deviation	2.6	2.5	2.4	2.5	2.3
Number of Stable Connections	17.06	16.6	21.6	21.7	22
Standard Deviation	2	1.6	2.1	2	2

highly-utilized WMN packet collisions happen quite frequently and in this situation TCP connections may constantly timeout for a long period of time. According to the design of AODV, if there is no traffic flowing on an established AODV routing path for a period of time (which is specified by the ACTIVE_ROUTE_TIMEOUT parameter), the source node will abandon the current path and re-flood the RREQ across the network to set up a new routing path. For this reason, when the TCP connection that uses the AODV routing path times out for a long period of time, AODV will abandon the used routing path and try to find a new one for the TCP connection. Flooding RREQ, however, consumes much network bandwidth and results in more packet collisions. In this condition, it is more difficult to establish routing paths and relay data packets. As a result, TCP connections more frequently trigger their retransmission timeouts and eventually become unstable.

The default the ACTIVE_ROUTE_TIMEOUT value is 3 seconds in AODV. As shown in Tab. 5.9, increasing the ACTIVE_ROUTE_TIMEOUT value can mitigate this problem and results in more stable TCP connections. However, although increasing the value of this parameter can generate more stable connections, it also causes AODV to respond more slowly to node movements. As will be presented later, this will result in a lower number of stable connections when mesh clients move in a WMN.

Tab. 5.10 shows the relationship between the hop count of the established connections and their achieved throughput in the OSPF routing protocol. As the average hop count of a connection decreases, the achieved throughput of the connection increases. This phenomenon shows that in a WMN "short" TCP connections usually can achieve more bandwidth than "long" TCP connections. We also studied the 20 runs of the OSPF case and found that if in a run there are more connections with fewer hop counts, the total download throughput of the run will be higher, as Fig. 5.15 shows. This phenomenon shows that using only the total download throughput as the sole performance metric to evaluate which routing protocol performs best may be misleading. A high total download throughput can be easily achieved by letting several "short" TCP connection monopolize

Table 5.10: Relationship between the average hop count and achieved throughput of connections

Hop Count	1	2	3	4
Throughput (KB/s)	14.33	11.19	9.12	7.6
Standard Deviation	3.2	3.1	3.7	2.7

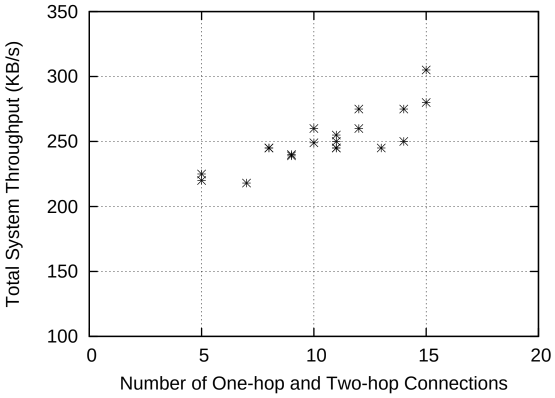


Figure 5.15: The relationship between the number of “short” connections and the total download throughput

the bandwidth of a WMN. Another important metric is the number of stable connections that can simultaneously exist in a WMN to share link bandwidth.

Multi-to-Multi Peer Traffic Configuration

In recent years, peer-to-peer applications are becoming more and more popular. Such applications include VoIP and music/video download applications. Here we created a simulation case to study the WMN performance with peer-to-peer applications. In this simulation case, each mesh client runs a TCP receiver (rtcp) and a TCP sender (stcp) and randomly sets up a greedy TCP connection to another mesh client. In total, there are 25 greedy TCP traffic flows in the system competing for the system bandwidth of the WMN. All mesh clients are fixed in this case.

Tab. 5.11 shows the total download throughput of the WMNs using this traffic pattern. As one sees, the achieved total download throughputs are higher than those reported in Tab. 5.7. Note that in the previous downlink TCP traffic case, all TCP traffic flows need to merge at the single gateway. For this reason, they are bottlenecked at a single point in the WMN. In this multi-to-multi peer traffic case, however, all TCP traffic flows need not merge at the single gateway. Instead, they can choose the best shortest routing paths

Table 5.11: The system total throughput in the multi-to-multi peer traffic case

Protocol	OSPF	STP	AODV
Total Download Throughput (KB/s)	756.9	569.8	603.1
Standard Deviation	32.2	22.3	25.6

Table 5.12: The number of established and stable connections in the multi-to-multi peer traffic case

Protocol	OSPF	STP	AODV
Number of Established Connections	21.68	20.25	16.94
Standard Deviation	2.3	2.2	1.5
Number of Stable Connections	11.79	5.8	8.56
Standard Deviation	2.1	0.6	1.6

in the WMN for relaying their data. The freedom of routing will improve the efficiency of wireless bandwidth usage in the WMN.

Tab. 5.11 shows that OSPF provides a higher download throughput than AODV and STP. The reason why OSPF outperforms STP is that in STP a routing path between two mesh clients may not be the shortest one because the routes determined by STP should follow the tree structure. As a result, the bandwidth of the WMN may not be efficiently utilized by STP. The reason why OSPF outperforms AODV has been explained before. In AODV, the active route timeout of a routing path is constantly triggered, which causes the RREQ to be flooded to the network, wasting the wireless bandwidth of the WMN.

Tab. 5.12 shows the number of established and stable connections of the WMNs with the three evaluated routing protocols. As can be seen, OSPF results in more stable connections than AODV and STP. The reason for AODV has been explained above. Here we explain the reason for STP. It is clear that, because packets can only be relayed on the spanning tree, STP may waste the network bandwidth due to the use of non-shortest-path route between a pair of mesh clients. Our simulation results show that a packet on average needs to traverse 3.99 hops to reach its destination client in STP while this number can be reduced to only 3.45 hops in OSPF. Due to this reason, given the same level of client traffic load, the level of congestion in STP is more severe than in OSPF. This means that more packets will be dropped in STP, which include the control packets of STP. As a result, the spanning tree constructed in STP may need to be constantly repaired or changed. However, this will cause TCP connections to timeout more often and make them unstable connections.

Table 5.13: The system total throughput under the mobility condition

Protocol	OSPF	STP	AODV
Total Download Throughput (KB/s)	243.2	244.1	230.8
Standard Deviation	14.8	14.9	13.2

Mobility Condition

In this section, we study the performance of the WMNs using the three evaluated routing protocols when mesh clients move. The settings of the used simulation case are the same as those of the downlink TCP traffic case, except that in the used case all mesh clients move at the speed of 1 m/s based on the random-waypoint mobility model. This model was first used by *Johnson and Maltz* in the evaluation of Dynamic Source Routing (DSR) [41], and was later refined by the same research group [42]. The refined version has become the de facto standard in mobile computing research.

As shown in Tab. 5.13, the WMNs using the three evaluated routing protocols achieve almost the same download throughputs. However, as can be seen in Tab. 5.14, AODV results in the least number of stable connections among the three evaluated routing protocols. This phenomenon is explained here. When a mesh client changes its associated mesh AP from the old AP to the new AP, its original routing path is no longer valid. Using AODV, however, the mesh APs on the original routing path need to wait for a long time before detecting such an event. Since the waiting time is long and a mesh client constantly changes its associated mesh AP while it moves, the disruption to a mesh client's connection is too long and too often, which makes many mesh clients' TCP connections unstable. In OSPF, when the new mesh AP gets the IEEE 802.11(b) association control packet from the mesh client, it broadcasts a LSA to inform other mesh APs that the mesh client now is associated with itself. If the old and new mesh APs are within each other's wireless transmission range, the LSA can reach the old mesh AP very quickly. This enables the old mesh AP to promptly forward the mesh client's packets to the new mesh AP and shortens the period of disruption to the mesh client. The details are shown in Fig. 5.16.

In STP, a similar mechanism is used to deal with node mobility. In STP, when the new mesh AP gets an association packet, it broadcasts a control packet up the spanning tree to inform upper-level mesh APs of this association change. When the control packet reaches an appropriate layer in the spanning tree, the packets destined to the mesh client

Table 5.14: The number of established and stable connections under the mobility condition

Protocol	OSPF	STP	AODV
Number of Established Connections	24.9	21.5	23.8
Standard Deviation	2.7	2.1	2.1
Number of Stable Connections	24.1	20.4	3.1
Standard Deviation	2.2	1.9	0.4

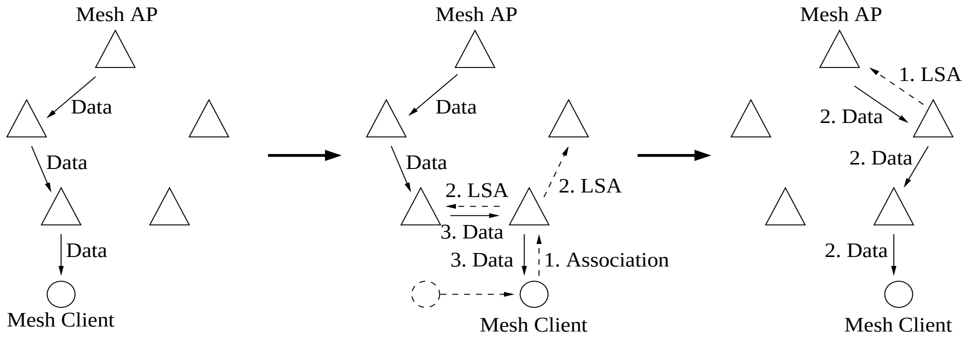


Figure 5.16: The handling of node movement in OSPF

will be directed toward the right branch of the spanning tree, which ends the period of disruption to the mesh client. This process is depicted in Fig. 5.17.

Multi-Gateway WMN

The total download throughputs of WMNs with a different number of mesh gateway nodes (i.e. those mesh APs connecting with the Internet gateway). These mesh gateway nodes are deployed at the corners of the simulated grid topology. Fig. 5.18 shows an example topology of a WMN with two mesh gateway nodes. OSPF is used in the cases in this section. The traffic settings of these cases are the same as those in the downlink TCP

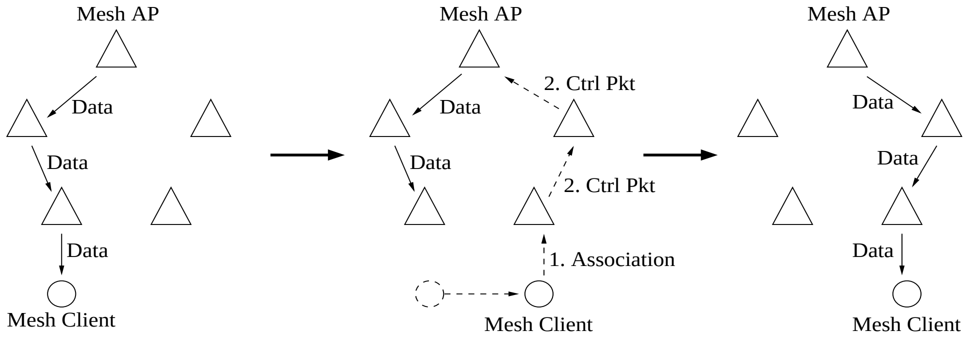


Figure 5.17: The handling of node movement in STP

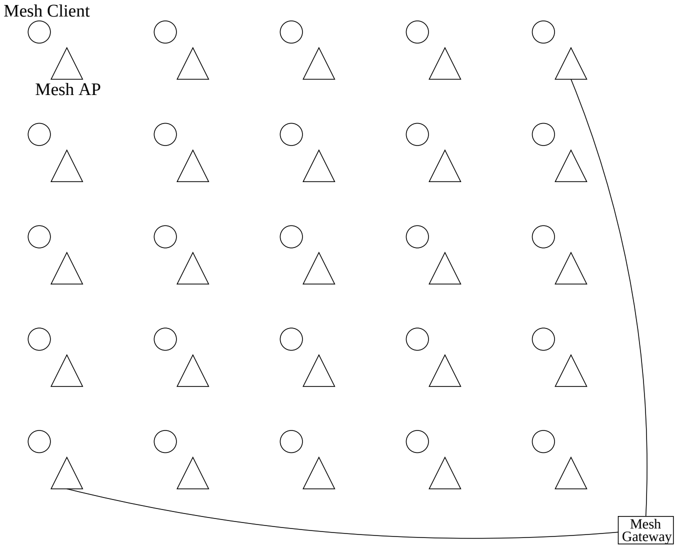


Figure 5.18: The network topology of a two-gateway 5x5 grid WMN

Table 5.15: The system total throughput of a multi-gateway WMN with different number of gateways

Number of Gateway	1	2	3	4
Total Download Throughput (KB/s)	254.1	381.7	670.4	903.1
Standard Deviation	15.5	17.6	28.4	39.6

traffic case, except that now multiple mesh APs connect to the mesh Internet gateway rather than just one. Tab. 5.15 shows the simulation results of these cases. As one sees, using more gateways in a WMN can significantly improve the total download throughput when most traffic in the WMN is Internet traffic. These results suggest that enough gateway nodes should be deployed in a WMN to make its performance scalable with the number of mesh clients.

OSPF with ETX

Here we study the effect of ETX when it is combined with the OSPF routing protocol. The traffic settings of the studied cases are the same as those of the downlink TCP traffic case. To let ETX show its capability in harsh wireless environments, the two-ray ground model with the Rayleigh fading [43] is used in this simulation case. The bit-error model for binary phase-shift keying (BPSK) modulation is adopted to calculate the bit error rate ($BER = 1/(2*(1+Power))$. For each received packet, its received power is calculated based on the distance between the source and destination nodes. The received power is

Table 5.16: The system total throughput under OSPF and OSPF with ETX in a harsh wireless environment

Protocol	OSPF with ETX	OSPF
Total Download Throughput (KB/s)	128.1	180.5
Standard Deviation	10.5	13.9

then added with a random fading with a variance of 10 dbm. Based on the resultant power, the Bit Error Rate (BER) for the received packet is calculated. The received packet is then dropped with a probability based on the calculated BER.

Tab. 5.16 shows that OSPF with ETX generates a lower total download throughput than OSPF in this harsh environment; however, Tab. 5.17 shows that OSPF with ETX allows more stable connections to coexist than OSPF in this harsh environment. The reason is that in this harsh environment the shortest paths selected by OSPF are usually with high BERs and thus fragile. In this condition, most TCP connections with high hop counts are broken (in the TCP timeout state) and contribute little to the total download throughput. As a result, the total download throughput is mostly contributed by those TCP connections with low (2 or 3) hop counts. That is, a few "short-distance" TCP connections (initiated by those mesh clients that are close to the gateway mesh APs) monopolize the network bandwidth and block out many "long-distance" TCP connections (initiated by those mesh clients that are far away from the gateway mesh APs). Because the RTTs and hop counts of these "short-distance" TCP connections are small, the TCP congestion control of these connections allow them to rapidly pump their data into the WMN (due to the well-known "self-clocking" property of TCP), even when they experience some packet losses.

In contrast, using the ETX metric in OSPF helps OSPF choose a low-BER and higher-throughput routing path for a connection. Thus, "long-distance" TCP connections become more robust and more of them can achieve a high throughput. However, the cost of this more even sharing of system bandwidth among "short-distance" and "long-distance" TCP connections is the reduced total download throughput. This is because the "long-distance" TCP connections in OSPF with ETX cannot react to packet losses as rapidly as the "short-distance" TCP connections in OSPF.

Table 5.17: The number of established and stable connections under OSPF and OSPF with ETX in a harsh wireless environment

Protocol	OSPF with ETX	OSPF
Number of Established Connections	24.75	24.8
Standard Deviation	3.1	2.9
Number of Stable Connections	22.7	6.75
Standard Deviation	3.1	2.3

Table 5.18: The total download throughput of single-radio and dual-radio WMNs

Protocol	OSPF (1c)	OSPF (2c)
Total Download Throughput (KB/s)	120.8	254.3
Standard Deviation	9.6	20.3

Single-Radio WMN vs. Dual-Radio WMN

One advantage of dual-radio-dual-mode WMNs over single-radio WMNs is that client-AP traffic and AP-AP traffic can be transported over different frequency channels at the same time to increase the total network throughput. To see how the second radio improves the total network throughput, we conducted two tests. In the first test, the ad-hoc mode and infrastructure mode interfaces of a mesh AP are set to use different channels. In contrast, in the second test, these two interfaces are set to use the same frequency channel. Because IEEE 802.11(b) MAC employs a carrier-sense multiple access (CSMA) mechanism to avoid collisions, multiple transmissions cannot be performed at the same time on the same channel. Thus, although the configuration of the WMN in the second test is not exactly the same as that of a single-radio WMN, their effects are the same: *client-AP traffic and AP-AP traffic cannot be transported at the same time in a WMN.*

Tab. 5.18 shows the total download throughputs achieved in the first and second tests when the OSPF routing protocol is used. The notation OSPF (1c) represents the case in which a mesh AP runs OSPF and its two interfaces are set to use the same channel. Similarly, OSPF (2c) represents the case in which a mesh AP runs OSPF and its two interfaces are set to use different channels. From this figure, one can see that using two radios can improve the total download throughput significantly.

5.4.8 Summary

In this section, we studied the performance of three classic routing protocols (AODV, STP, and OSPF) operating in dual-radio-dual-mode infrastructure WMNs. Each of these protocols represents a different and popular approach that has been adopted by some commercial WMN products. AODV represents the ad hoc routing protocols developed for mobile ad hoc networks; STP represents the traditional routing approach developed for fixed bridges/switches; and OSPF represents the traditional routing protocols developed for the Internet. Due to the importance of these protocols, in this section they are selected for studying and evaluating the effects of routing protocols on dual-radio-dual-mode WMNs.

Our simulation results show that OSPF outperforms the others in dual-radio-dual-mode WMNs. Compared with AODV and STP, it provides higher total download throughputs, allows more stable connections to coexist in a WMN, and responds more quickly to the movement of mesh clients. Our simulation results show that with the ETX metric support, the total download throughput of a WMN can be more evenly shared by mesh clients rather than monopolized by a few mesh clients. We also implemented and studied the performance of multi-gateway WMNs. Our simulation results show that, when traffic in a WMN are mostly Internet traffic, a multi-gateway WMN can provide a much higher total download throughput than a single-gateway WMN. This suggests that one should deploy enough mesh gateways to make a WMN scalable with the number of mesh clients.

Chapter 6

Related Work

Very few papers in the literature have studied the performances of the distributed coordinated scheduling mode of the IEEE 802.16 mesh network. Most papers (e.g., [44], [45], [46]) focus on the centralized scheduling mode. For the distributed coordinated scheduling mode, the authors in [2] study the effect of the holdoff time value in terms of control message transmission cycles and the time required for establishing data schedules. The authors vary the holdoff time exponent value from 0 to 4 to observe the effects of the resultant holdoff time value. In [3], the authors analyze control message transmission cycles and round trip times (RTT) using different holdoff time base values.

In [21], Bayer et al. propose a dynamic holdoff time setting scheme to improve the scheduling performance of the distributed scheduling mode. The goal of this paper is similar to that of ours. However, there are fundamental design and implementation differences between them. In the following, we present some details of this paper to point out the differences.

The main idea of the scheme proposed in [21] is as follows. Network nodes that are transmitting, receiving, or forwarding data packets should use smaller holdoff time values to exchange MSH-DSCH messages more quickly. In contrast, nodes that are not transmitting, receiving, or forwarding data packets should use larger holdoff time values to reduce contentions for TxOpps. The authors classify network nodes into four classes: BS, active, sponsoring, and inactive. The BS class comprises all BS nodes in the network; the sponsoring class consists of nodes that are sponsoring new nodes (i.e., allocating data schedules for forwarding the control messages initiated from a new node's network registration procedure). A sponsoring node is a node that has been selected by one of

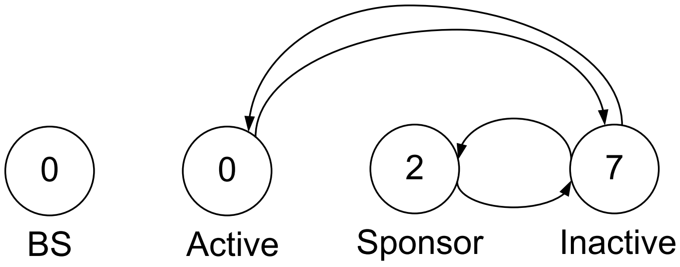


Figure 6.1: The maximum holdoff exponent values of the four classes proposed in [21] and the transitions among them

its neighbors as next hop towards the destination. Network nodes that are transmitting, receiving, or forwarding data packets are called active nodes; nodes that are idle are called inactive nodes. Each of these four classes has its own range for the holdoff time exponent value. The upper bounds of these classes' holdoff time exponent values are shown as follows:

$$0 \leq ME_{BS} \leq ME_{act} < ME_{sp} < ME_{inact} \leq 7 \tag{6.1}$$

where ME_{BS} denotes the maximum holdoff time exponent value of the BS class; ME_{act} denotes the maximum holdoff time exponent value of the active class; ME_{sp} denotes the maximum holdoff time exponent value of the sponsoring class; and ME_{inact} denotes the maximum holdoff time exponent value of the inactive class. We plot the maximum holdoff exponent values of the four classes used in [21] and the transitions among them in Fig. 6.1.

Since a smaller holdoff time exponent value results in a smaller holdoff time value, nodes using a smaller holdoff time exponent value will refrain themselves from contending for TxOpps for a shorter period of time. Such nodes, therefore, can on average win a TxOpp faster than nodes using a larger holdoff time exponent value. Due to this reason, nodes of the BS class on average can transmit MSH-DSCH messages faster than (or as fast as, if $ME_{BS} = ME_{act}$) the nodes belonging to the active class. Similarly, nodes belonging to the active class on average can transmit MSH-DSCH messages faster than the nodes belonging to the sponsoring or inactive class.

Although this scheme provides some advantages, it has several disadvantages described here. First, this scheme needs to rely on a collaborative routing protocol to determine if a node should belong to the active class or the inactive class. For each node, it should consult the routing protocol to check whether it is selected as a potential forwarding node for a routing path. If so, it should switch to the active class. Otherwise, it should belong to the inactive class.

Second, when a mesh network becomes highly-loaded, it is very likely that every node has data to send most of the time. In this condition, most of the nodes will belong to the active class and thus will have the same range of the holdoff time value. Thus, the multi-class design of this scheme will degenerate to the original single-class design in this condition. For example, if all nodes in the network have data to send or receive, all of them will set their holdoff time exponent value to zero based on the design proposed in [21]. This means that in this condition this scheme degenerates to the HT-1 scheme. To avoid this problem, which is caused by heavy load, an admission control mechanism should be used.

Third, actually not all active nodes need to use smaller holdoff time values at all time. Active nodes need not establish data schedules over all TxOpps that they win. This is because an established data schedule can be valid for N frames, where $1 \leq N \leq 128$. (Note that each frame can contain M TxOpps, where $2 \leq M \leq 15$, depending on the network setting.) If an active node excessively wins transmission opportunities without considering whether it has data to send, it will waste many TxOpps that could otherwise be given to other nodes that have data to send.

Lastly, in this scheme, when switching to a higher-priority class, a node should first set its holdoff time exponent value to zero and then gradually increment this value by one until its and its neighboring nodes' Mx values are no longer above a pre-defined threshold. This means that before a node can stabilize its holdoff time exponent value, it will excessively contend for TxOpps and thus waste the control-plane bandwidth. To understand the meaning of Mx, the standard defines that each node should use two shorter fixed-length fields, *exp* and *Mx*, to represent its next transmission opportunity number. The relationship between these two fields and a TxOpp number is given in (3.3). The detailed information about (3.3) will be explained in Section 3.1.3.

In contrast, our proposed two-phase holdoff time setting scheme does not incur the above problems. Besides improving the scheduling performances of the distributed coordinated mode, our proposed scheme ensures the success of network initialization without wasting the control-plane bandwidth. The dynamic approach of our proposed scheme can dynamically reduce a node's holdoff time exponent value when the node needs to

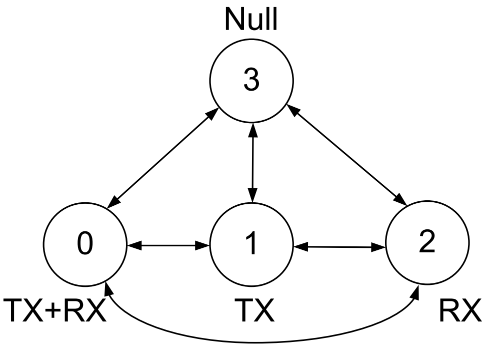


Figure 6.2: The holdoff time setting algorithm proposed in [5]

establish a data schedule. Thus, the dynamic approach can use TxOpps more efficiently than the scheme proposed in [21]. In addition, the dynamic approach need not employ a collaborative routing protocol; therefore, its design and implementation complexities are much lower than those of the scheme proposed in [21].

In [5], Kim et al. proposed an adaptive holdoff time algorithm based on the statuses of nodes. As shown in Fig. 6.2, in this scheme each node first sets its holdoff time exponent value to 3, which is the maximum holdoff time exponent value used in this work. Within a pre-defined interval, a node is classified into four statuses: 1) Null; 2) TX; 3) RX; and 4) TX+RX. The Null status means that the node is idle; the TX and RX statuses mean that the node has data to send and receive, respectively; and the TX+RX status means that the node has data to send and receive within this interval. The holdoff time algorithm proposed in this work has two problems. First, if all nodes have data to send and receive in a period, the scheme will set the holdoff time exponent values of all nodes to zero, which means that in this condition it will degenerate to the HT-1 scheme. Second, in the TX and RX statuses a node decreases its holdoff time exponent value with fixed values. This means that this scheme does not accurately compute how the holdoff time of a node can be set to minimize the time required for completing THPs. As a result, the network performance under this scheme can be suboptimal. In contrast, our proposed dynamic holdoff time scheme can reduce the time required for a node to complete a THP as much as possible. Thus, the performance of a network using our proposed dynamic holdoff time scheme can be nearly optimal.

6.1 Regarding Wireless Networks with Directional Antennas

In the literature, many papers have studied the issues and performances on using directional antennas in the 802.16 Point-to-MultiPoint (PMP) mode and the 802.16(d) mesh Centralized Scheduling (CS) mode. However, both of these standards greatly differ from the 802.16(d) mesh CDS mode in control message and data schedulings. In these two network standards, control message and data transmissions are scheduled on a central node using a tree-based network topology. The control message scheduling is using a centralized approach rather than a distributed control algorithms like MEA. Thus, to save space we do not list them in this section. In [47], *Bhatnagar et al.* discussed the deployment cost of an IEEE 802.16-based backhaul network, in which nodes are assumed to be equipped with directional antennas. They computed the minimum cost to deploy an IEEE 802.16-based network using integer programming. However, this paper does not address the operational issues of an 802.16(d) mesh CDS-mode network using single-switched-beam antennas and does not present numerical network capacity results.

Numerous papers have studied the issues and performances of using a variety of directional antennas in *Carrier Sense Multiple Access/Collision Avoidance* (CSMA/CA) networks [48, 49, 50, 51, 52, 53, 54, 55]. However, proposals in these papers operate on CSMA/CA networks rather than TDMA networks. Because the medium access schemes of these two types of networks greatly differ, these works greatly differ from ours.

Another track of the directional antenna research is based on TDMA-like networks. In [11], Deopura et al. proposed a centralized algorithm based TDMA network using smart antennas to provide link-layer differentiated service. In this work, a node is designated to be the central scheduler which is mainly responsible for receiving other nodes' traffic requests and performing the proposed centralized scheduling algorithm to determine which links should be activated on each time slot. This work requires a centralized algorithm to schedule data transmission in a per-slot manner and a data transmission is negotiated using a two-phase "request-reply" procedure, while our work uses a 802.16(d) CDS-mode based distributed exponential holdoff algorithm and a pair of node should negotiate a data transmission using THP.

In [12], Sundaresan et al. proposed both a centralized and distributed algorithms that exploit the degree of freedom (DOF) capability of adaptive antennas to increase the concurrency of data transmissions. The authors consider the DOF scheduling problem as a graph coloring problem with additional constraints and assume that the proposed algorithms are executed over TDMA networks. However, the effects of control message transmissions and the overheads due to the data scheduling handshake protocol were not studied in this work. Hence, this work differs from ours.

In [7], Sundaresan and Sivakumar proposed a generic MAC-layer framework for scheduling data transmissions in wireless ad-hoc networks with three-types of smart antennas (i.e., switched-beam antennas, adaptive array antenna, and MIMO array antennas). The proposed framework considers such a scheduling problem as an optimization problem formulated with node graphs, flow contention graphs, and resource constraint graphs. The generations of these graphs may differ depending the characteristics of the formulated antenna technology. The distributed algorithm proposed in [7] is based on the Proportional Fair Contention Protocol (PFCP) [56], which is a CSMA/CA-like medium access scheme with slotted link bandwidth. Thus, the assumptions used in [7] differs from those used in our work.

Several collision-free distributed algorithms based on time-division networks have been proposed. In [13], Singh et al. proposed a MAC-layer protocol that uses adaptive array antenna and the information of Directional of Arrival (DoA) to maximize the SINR value on receiving nodes in slotted ALOHA networks. The proposed protocol uses a "tone-data-ack" transmission sequence for slotted aloha networks. In this protocol, before transmitting data the transmitting node first directionally sends a "tone" special symbol towards the intended receiving nodes. All nodes that are idle (without transmitting/receiving data) should receive incoming signals omnidirectionally. After receiving a complex sum of all such tone signals, an idle node should run a DoA algorithm to identify the direction and strength of the incoming signals. It then forms a beam in the direction of the detected signal with the maximum strength and nulls in other directions, for receiving data packets sent by the transmitting node and cancels interferences from other directions. After receiving the data, if it is the destination of the received data, it should send the transmitting node an ACK message using the same beam-forming setting. Otherwise, it should discard the data. Because our work need not require nodes to receive messages omnidirectionally and schedules data transmissions using a three-way approach rather than the two-way “omnidirectional-tone and directional-data” pattern, this work differs from ours.

In [14][15], Bao and Garcia-Luna-Aceves proposed a receiver-oriented multiple access algorithm (called ROMA) for ad-hoc networks with directional antennas. ROMA uses a hash function to determine whether a node can transmit data or receive data in each time slot. In ROMA, control messages are omnidirectionally exchanged and the data transmission/reception of a node is not determined in an on-demand manner. Thus, the protocol behaviors of ROMA greatly differ from those of our proposed 802.16(d) mesh CDS-mode network using pure directional transmissions.

In [16][17], Steenstrup proposed two MAC-layer protocols for exploiting directional antennas in TDMA networks. In [16], each node has a separate omnidirectional channel to exchange control messages. Over this omnidirectional channel, a pair of neighboring nodes should first establish a negotiation slot for exchange their control messages using a two-way procedure. In this procedure, the requesting node first sends a message containing a list of preferred time slots that can be used for exchanging data with the replying node (called negotiation slots). Upon receiving this message, the replying should first choose one time slot from the received slot list for communicating with the requesting node and then send a response message containing its chosen time slot number back to the requesting node. Later on, they can use the chosen negotiation slots to perform a “request-grant-confirm” three-way handshake procedure to schedule a data transmission with directional antennas.

In [17], each node is required to perform omnidirectional reception when being idle. A time frame is divided into three segments: negotiation segment, reservation segment, and contention segment. Each segment comprises several smaller time slots. Before transmitting data, a transmitting node should first randomly choose a time slot in the negotiation segment to directionally send the receiving node a request message containing the number of requested time slots and the set of available time slots in the reservation segment to transmit data. On omnidirectionally receiving such a request message, the receiving node replies a response message to the transmitting node to indicate whether it can receive data from the transmitting node and on which slots in the reservation segment it can receive data from the transmitting node. If a feasible schedule is negotiated, the transmitting node will transmit data to the receiving node on the chosen time slots in the reservation segment using directional transmissions and receptions. Time slots in the contention segment are used by nodes for data retransmissions in case data losses are detected. Because our work need not use omnidirectional receptions for management the network and control messages are not exchanged in a random manner, our work greatly differs from [16][17].

A MAC protocol that uses pure directional transmissions/receptions (called DTRA) has been proposed by Zhang et al. in [8][9][10]. In DTRA, each node is assumed to be equipped with directional antennas. A time frame is divided into three segments: the neighbor discovery segment, the reservation segment, and the data transmission segment. Each segment is further divided into several slots. When entering the network, each node should perform a scanning procedure to probe/learn possible neighboring nodes. In this scanning procedure, each node should determine whether it will actively transmit advertisement messages or passively listen any advertisement messages in a "scan." A scan is defined as a sequence of antenna-pointing directions and mode selection (i.e., the transmitting and listening mode) on slots in the neighbor discovery segment. The mode selection is determined by the values of the binary digits of each node's identifier. Given that each node is assigned an identifier in a binary form. In the i-th scan, a node will actively transmit advertisement messages if the i-th digit of its identifier is '1'. Otherwise, it should passively listen incoming messages.

During the scanning procedure, each node also has to perform a three-way handshake procedure (denoted as THP-1) with possible neighboring nodes via its sent advertisement message. In this three-way handshake procedure, a pair of nodes exchange their free slots in the reservation segment and negotiate several common slots in the reservation segment for scheduling future data transmission. The data transmission schedule of a pair of nodes should be negotiated using another three-way handshake procedure (denoted as THP-2) to obtain a set of free slots that are accepted by the two nodes in the data transmission segment. For a pair of nodes, control messages of their THP-2 are transmitted on the slots in the reservation segment negotiated by their THP-1. DTRA uses a three-way handshaking approach for nodes to schedule their control messages used for scheduling data transmissions. Thus, the timings of those control message transmissions can be arbitrary. The objective of our work is to study the performances of the IEEE 802.16(d) mesh CDS-mode network with single-switched-beam antennas, which uses an exponential holdoff time design to control each node's control message transmission. Due to the great distinction between the control message schedulings of the two networks, the behaviors of DTRA differ from those of our studied networks.

In addition to the distinction of the operations of the studied networks, in this dissertation we also studied the performances of a real-life TCP implementation (BIC-TCP) over IEEE 802.16(d) mesh CDS-mode network with single-switched-beam antennas. To the best of the authors' knowledge, none of the papers in the literature have studied the real-life TCP performances over such new type of networks with single-switched-beam antennas.

6.2 Regarding IEEE 802.16(d) Mesh CDS-mode Networks

The effects of a node's holdoff time value on IEEE 802.16(d) mesh CDS-mode networks with omnidirectional radios have been extensively studied. In [2], Cao et al. built a stochastic model to analyze the scheduling performance of the IEEE 802.16(d) mesh CDS mode. Using the built model, the authors studied the scheduling performances of the CDS mode under several different system parameter values. In [3], Bayer et al. studied the transmission intervals of control messages and the round trip times (RTT) of data packets under several different system parameter values. Based on these findings, several proposals that allow nodes to use dynamic holdoff times were proposed to enhance the performances of this new type of networks [4][21][5][57][58][6]. The main idea of these dynamic holdoff time schemes are to dynamically adjust each node's holdoff time value according to several criteria (e.g., its output buffer occupancy, its intention to send data, etc.) to increase network performances without generating network congestion. For brevity, rather than presenting the details of these works, we only point the key differences between these works and ours. The works in [2]-[6] assume that nodes in an 802.16(d) mesh CDS-mode network are equipped with omni-directional antennas and thus each node needs to run one instance of MEA to schedule control messages. However, our work considers an 802.16(d) mesh CDS-mode network where each node is equipped with a single-switched-beam antenna. Due to limited coverage of such an antenna, each node is required to run multiple instances of MEAs with the capabilities of using dynamic holdoff times and eliminating scheduling conflicts among other MEAs on the same node. For these differences, the objective and algorithm of our proposed dynamic holdoff time scheme greatly differ from those of the previous works.

Regarding the scheduling of data transmissions, three variants of THP have been proposed for the 802.16(d) mesh CDS mode [59][60][61]. In [59], Lin et al. proposed an extension to THP that allows the granting node to grant a minislot allocation that contains fewer minislots than that the requesting node requests, if the network is congested, to improve the scheduling efficiency of a THP. In [60], Wang et al. considered the scheduling conflict problems resulting from two THPs that are run simultaneously on neighboring nodes. In their proposal, when a node detects a conflict between the minislot allocation scheduled by its ongoing THP and that scheduled by its neighboring node, it should first cancel the transmission of the confirm IE of its ongoing THP and then send a request IE to its granting node to indicate the minislot allocation just scheduled is no longer valid.

In [61], Wang et al. first observed that the Quality-of-Service (QoS) requirement of a flow is associated with a MAC-layer connection. However, the THP used in the standard schedules a minislot allocation for a link between two nodes. Multiple MAC-layer connections can be mapped on the same link. Because the control messages used in THP cannot carry any QoS-related information, the granting node of a THP cannot reserve minislots for connections based on their QoS requirement, greatly reducing the capability of the 802.16(d) mesh CDS-mode to support QoS. To solve this problem, the authors introduces a new QoS IE into THP so that the granting node can know the priority and the QoS requirement associated with the bandwidth request when scheduling minislot allocations. Based on this new design, three THP variants that allows nodes to transmit multiple request and grant IEs were proposed to enhance the scheduling efficiency of THP. In this dissertation, we use the original THP design to evaluate the performances of the 802.16(d) mesh CDS mode with single-switched-beam antennas. To exploit the spatial reuse property of single-switched-beam antennas, a transmission-domain-aware minislot scheduler was developed in our work. However, the new THP designs proposed in [59][60][61] can be used in our work to further improve the protocol efficiency of the THP itself.

In [62], Xiong et al. observed that in an 802.16(d) mesh CDS-mode network with omnidirectional antennas two neighboring nodes can simultaneously transmit their data to different nodes as long as the receiving node of a transmission-reception pair will not be interfered by the transmitting node of the other transmission-reception pair. With this relaxation in data scheduling, the concurrency of activated links can be increased, as compared with a conservative data scheduler that does not allow such concurrent transmissions. In our implementation, such a scheduling relaxation has been realized in the minislot scheduler. In addition, to exploit spatial reuse advantage of single-switched-beam antennas, we developed a transmission-domain-aware minislot scheduler that can utilize the advantage of single-switched-beam antennas to increase the data transmission concurrency of the 802.16(d) mesh CDS-mode network.

Chapter 7

Conclusion

7.1 Final Remarks

In this dissertation, we propose several schemes to enhance the scheduling efficiency of the IEEE 802.16 mesh CDS-mode network on the control plane and the QoS support of this new network on the data plane. We also point out the issues of network initialization of this network on random topologies and propose a scheme to solve these issues. The performances of our proposed schemes are evaluated using both analyses and simulations. Both of the analytical and numerical results show that our proposed schemes can significantly enhance the performance of the IEEE 802.16 mesh CDS-mode network and benefits upper-layer applications. In addition to MAC-layer and application performance, we also study the impacts of three classic routing protocols on WMNs using simulations. Our simulation results suggest that, as compared with the AODV and STP protocols, OSPF is most suitable for WMNs.

7.2 Future Work

The proposed dynamic holdoff time scheme can be further extended from several aspects, which are discussed below.

7.2.1 Power Saving

The main contribution of our proposed dynamic holdoff time scheme is to decrease the time required for nodes to establish minislot allocations by dynamically changing nodes' holdoff times based on the dynamic needs of traffic flows. Decreasing the time required for establishing minislot allocations shortens the idle time of the MAC layer. Thus, using our proposed schemes the throughputs of applications can be significantly increased and the end-to-end delay of applications can be significantly decreased.

The objectives of our proposed dynamic holdoff time schemes, however, are unnecessary to be restricted to the performance aspect only. The proposed dynamic holdoff time schemes can be extended to reduce the power consumption of nodes in an IEEE 802.16 mesh CDS-mode network. Due to the foreseeable energy deficiency in the future, devices and equipments are required to be power-efficient, even though they are allowed to use power-line infrastructure. For this reason, it is valuable to reduce the power consumption of an IEEE 802.16 mesh CDS-mode relay node. This can be accomplished by extending our proposed dynamic holdoff time scheme.

On one hand, each node can set its holdoff time to a large value, when it does not have data to send. Doing so will increase the number of idle TxOpps, which allows a node to have more chances to turn off its radio. On the other hand, to fast recover from the sleeping state (and to fast respond to user's request), each node cannot set its holdoff time to a too large value to prevent itself from sleeping for a too long period. These two contradictory factors makes designing a dynamic holdoff time scheme for power-saving complicated and worth further studying.

7.2.2 QoS Support

In Section 5.3, we point out the QoS support flaw in THP and adds a new IE called QoS IE to fix this flaw. The refined THP architecture has been validated using simulations. Our simulation results show that under this architecture a pair of nodes can schedule minislot allocations at the same time for serving traffic flows with different QoS requirements. Based on this refined THP architecture, one can design and implement advanced schemes and protocols for serving multi-media and real-time traffic in this new WMN.

7.2.3 Mathematic Modeling

In Section 4.3, we evaluate the performance of the proposed dynamic holdoff time scheme in networks using single-switched-beam antennas. Numerical results have shown the effectiveness of the proposed dynamic holdoff time scheme in such networks. To the best of our knowledge, so far no papers model the complex behaviors of IEEE 802.16 mesh CDS-mode networks using single-switched-beam antennas. Due to the complexity of such networks, studies modeling this new network with single-switched-beam antennas are valuable to obtain more insights.

Bibliography

[1] IEEE Std 802.16 2004 (Revision of IEEE Std 802.16-2001), *IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems*, 2004.

[2] M. Cao, W. Ma, Q. Zhang, X. Wang, and W. Zhu, "Modelling and performance analysis of the distributed scheduler in ieee 802.16 mesh mode," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM MobiHoc 2005)*, vol. 1, pp. 165–168, Urbana-Champaign, IL, USA, May 25-27, 2005.

[3] N. Bayer, D. Sivchenko, B. Xu, V. Rakocevic, and J. Habermann, "Transmission timing of signaling messages in ieee 802.16 based mesh networks," in *Proceedings of the 12th European Wireless Conference (EW 2006)*, vol. 1, pp. 165–168, Athens, Greece, April 2-5, 2006.

[4] S.Y. Wang, C.C. Lin, H.W. Chu, T.W. Hsu, and K.H. Fang, "Improving the Performances of Distributed Coordinated Scheduling in IEEE 802.16 Mesh Networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 4, July 2008.

[5] B. C. Kim, D. G. Kwak, H. Song, H. S. Lee, and J. S. Ma, "An adaptive holdoff algorithm based on node state for ieee 802.16 mesh mode with coordinated distributed scheduling," in *Proceedings of the 19th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2008)*, vol. 1, pp. 1–5, Cannes, France, Sept. 15–18, 2008.

[6] Y. Li, D. Wei, H. Zhuang, H. Wang, and P. Wang, "A new congestion control method for ieee 802.16 mesh mode," in *Proceedings of the 2009 International Conference on*

Communication Software and Networks (ICCSN 2009), vol. 1, pp. 726–730, Chengdu Sichuan, China, Feb 2009.

[7] K. Sundaresan and R. Sivakumar, "A unified mac layer framework for ad-hoc networks with smart antennas," *IEEE/ACM Transactions on Networking*, vol. 15, no. 3, pp. 546–559, June 2007.

[8] Z. Zhang, "Pure directional transmission and reception algorithms in wireless ad hoc networks with directional antennas," in *Proceedings of the IEEE International Conference on Communications 2005, (ICC 2005)*, vol. 5, pp. 3386–3390, Seoul, Korea, May 16–20, 2005.

[9] Z. Zhang, B. Ryu, G. Nallamothu, and Z. Huang, "Performance of all-directional transmission and reception algorithms in wireless ad hoc networks with directional antennas," in *Proceedings of the IEEE Military Communications Conference 2005 (MILCOM 2005)*, vol. 1, pp. 225–230, Atlantic City, NJ, USA, Oct. 17–20, 2005.

[10] Z. Zhang, "Dtra: Directional transmission and reception algorithms in wlans with directional antennas for qos support," *IEEE Network*, vol. 19, no. 3, pp. 27–32, May-June 2005.

[11] A. Deopura and A. Ganz, "Provisioning link layer proportional service differentiation in wireless networks with smart antennas," *Wireless Networks*, vol. 13, no. 3, pp. 371–378, June 2007.

[12] K. Sundaresan, W. Wang, and S. Eidenbenz, "Algorithmic aspects of communication in ad-hoc networks with smart antennas," in *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2006)*, vol. 1, pp. 298–309, Florence, Italy, May 22-25 2006.

[13] H. Singh and S. Singh, "Smart-aloha for multi-hop wireless networks," *Mobile Networks and Applications*, vol. 10, no. 5, pp. 651–662, October 2005.

[14] L. Bao and J. J. Garcia-Luna-Aceves, "Transmission scheduling in ad hoc networks with directional antennas," in *Proceedings of the 8th annual international conference on Mobile Computing and Networking (MobiCom 2002)*, pp. 48–58, Atlanta, Georgia, USA, Sep. 23-28, 2002.

[15] L. Bao and J. J. Garcia-Luna-Aceves, “Receiver-oriented multiple access in ad hoc networks with directional antennas,” *Wireless Networks*, vol. 11, no. 1-2, pp. 67–79, January 2005.

[16] J. Cain, T. Billhartz, L. Foore, E. Althouse, and J. Schlorff, “A link scheduling and ad hoc networking approach using directional antennas,” in *Proceedings of the IEEE Military Communications Conference 2003 (MILCOM 2003)*, vol. 1, pp. 643–648, Boston, MA, USA, Oct. 13-16, 2003.

[17] M. Steenstrup, “Exploiting directional antennas to provide quality of service and multipoint delivery in mobile wireless networks,” in *Proceedings of the IEEE Military Communications Conference 2003 (MILCOM 2003)*, vol. 2, pp. 987–992, Boston, MA, USA, Oct. 13-16, 2003.

[18] J. N. Daigle, *Queueing Theory with Applications to Packet Telecommunication*, ch. 5, p. 214. Springer, 1 ed., 2005.

[19] S.Y. Wang, C.L. Chou, C.H. Huang, C.C. Hwang, Z.M. Yang, C.C. Chiou, and C.C. Lin, “The Design and Implementation of the NCTUns 1.0 Network Simulator,” *Computer Networks*, vol. 42, no. 2, pp. 175–197, June 2003. (available at http://NSL.csie.nctu.edu.tw/nctuns.html).

[20] C. A. Balanis, *“Antenna Theory Analysis and Design (2nd Edition)”*. Chap. 6, Wiley, New York, 1997.

[21] N. Bayer, B. Xu, J. Habermann, and V. Rakocevic, “Improving the performance of the distributed scheduler in ieee 802.16 mesh networks,” in *Proceedings of the IEEE 65th Vehicular Technology Conference (VTC 2007-Spring)*, Dublin, Ireland, April 22-25, 2007.

[22] S.Y. Wang, C.C. Lin, K.H. Fang, and T.W. Hsu, “Facilitating the Network Entry and Link Establishment Processes of IEEE 802.16 Mesh Networks,” in *Proceedings of the IEEE Wireless Communications and Networking Conference 2007 (WCNC 2007)*, Hong Kong, March 11-15 2007.

[23] I. F. Akyildiz, X. Wang, and W. Wang, “Wireless mesh networks: A survey,” *Computer Networks*, vol. 47, no. 4, pp. 445–487, March 15 2005.

[24] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proceedings of the 10th ACM Annual International Conference on Mobile Computing and Networking (MobiCom 2004)*, Philadelphia, PA, USA, Sep. 26-Oct. 1 2004.

[25] BelAir Networks, "Mesh Networking Options," available from http://www.belairnetworks.com/about_belair/challenge_mesh.cfm/.

[26] "MeshDynamics Structured Mesh," available from http://www.meshdynamics.com/.

[27] E. Royer and C. Toh, "A review of current routing protocols for ad-hoc mobile wireless networks," *IEEE Personal Communication Magazine*, vol. 6, no. 2, no. 2, pp. 46–55, 1999.

[28] Microsoft Research, "Mesh Networking," available from http://research.microsoft.com/mesh/.

[29] D. Aguayo, J. Bicket, S. Biswas, D. D. Couto, and R. Morris, "MIT Roofnet Implementation," available from http://pdos.lcs.mit.edu/roofnet/design/.

[30] D. Maltz, D. Johnson, and Y. Hu, "The dynamic source routing protocol for mobile ad hoc networks (dsr)," *Internet Draft*, April 2003.

[31] D. Maltz, D. Johnson, and Y. Hu, "Dynamic source routing in ad hoc wireless networks," *Internet Draft*, December 1998.

[32] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," *IETF RFC 3561*, July 2003.

[33] J. Moy, "Ospf version 2," *IETF RFC 2328*, April 1998.

[34] "Tropos Networks," available from http://www.tropos.com/technology/whitepaper.html/.

[35] ANSI/IEEE Std 802.1D, *"IEEE 802.1D Specification"*, 1998.

[36] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proceedings of the 1994 ACM SIGCOMM Conference (SIGCOMM 1994)*, pp. 234–244, London, UK, Aug. 31-Sep. 2 1994.

[37] IEEE, *IEEE Std. 802.11-1999. Part II: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 1999.

[38] G. Malkin, *RIP Version 2.* IETF RFC 2453, November, 1998.

[39] ANSI/IEEE Std 802.1, *IEEE 802.1*, 1998.

[40] K. Fall and K. Varadhan, "The *ns* Manual," http://www.isi.edu/nsnam/.

[41] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Chap. 5, Mobile Computing*, pp. 153–181, Imielinski and Korth, Eds. Kluwer Academic Publishers, 1996.

[42] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of the 1998 ACM Annual International Conference on Mobile Computing and Networking (MOBICOM 1998)*, pp. 85–97, Dallas, Texas, USA, Oct. 25-30 1998.

[43] B. Sklar, "Rayleigh fading channels in mobile digital communication systems part i: Characterization," *IEEE Communications Magazine*, vol. 35, no. 7, pp. 90–100, July 1997.

[44] M. Cao, V. Raghunathan, and P. R. Kumar, "A tractable algorithm for fair and efficient uplink scheduling of multi-hop wimax mesh networks," in *Proceedings of the 2nd IEEE Workshop on Wireless Mesh Networks (WiMesh 2006)*, pp. 101–108, Reston, VA, USA, Sep. 25 2006.

[45] D. Kim and A. Ganz, "Fair and efficient multihop scheduling algorithm for ieee 802.16 bwa systems," in *Proceedings of the 2nd IEEE International Conference on Broadband Networks (Broadnets 2005)*, Boston, USA, Oct. 3-7, 2005.

[46] H. Shetiya and V. Sharma, "Algorithms for routing and centralized scheduling to provide qos in ieee 802.16 mesh networks," in *Proceedings of the 1st ACM Workshop on Wireless Multimedia Networking and Performance Modeling (ACM WMuNeP 2005)*, Montreal, Canada, Oct. 13, 2005.

[47] S. Bhatnagar, S. Ganguly, and R. Izmailov, "Design of IEEE 802.16-based Multi-hop Wireless Backhaul Networks," in *Proceedings of the 1st International Conference on Access Networks (AcessNets 2006)*, p. 5, ACM Press, Athens, Greece, Sep. 4-6, 2006.

[48] Y. Huang, W. Gong, and D. Gupta, “MCMSDA: A Multi-Channel Multi-Sector Directional Antenna Wireless LAN,” in *Proceedings of the 2006 IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks (WOWMOM 2006)*, pp. 49–58, IEEE Computer Society, Buffalo, NY, USA, June 26-29 2006.

[49] M. Takai, J. Martin, R. Bagrodia, and A. Ren, “Directional virtual carrier sensing for directional antennas in mobile ad hoc networks,” in *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking Computing (MobiHoc 2002)*, pp. 183–193, ACM Press, Lausanne, Switzerland, June 9-11, 2002.

[50] R. R. Choudhury, X. Yang, N. H. Vaidya, and R. Ramanathan, “Using directional antennas for medium access control in ad hoc networks,” in *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking (MobiCom 2002)*, pp. 59–70, ACM Press, Atlanta, Georgia, USA, Sep. 23-28, 2002.

[51] R. Ramanathan, J. Redi, C. Santivanez, D. Wiggins, and S. Polit, “Ad hoc networking with directional antennas: a complete system solution,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 3, pp. 496–506, March 2005.

[52] Y. B. Ko, V. Shankarkumar, and N. Vaidya, “Medium access control protocols using directional antennas in adhoc networks,” in *Proceedings of the 19th IEEE Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2000)*, pp. 13–21, Tel-Aviv, Israel, March 26-30, 2000.

[53] M. Neufeld and D. Grunwald, “Using phase array antennas with the 802.11 mac protocol,” in *Proceedings of the First International Conference on Broadband Networks 2004 (BroadNets 2004)*, pp. 733–735, San Jose, California, USA, Oct. 25-29, 2004.

[54] R. Choudhury and N. H. Vaidya, “Deafness: A MAC Problem in Ad Hoc Networks when using Directional Antennas,” in *Proceedings of the 12th IEEE International Conference on Network Protocols 2004 (ICNP 2004)*, pp. 283–292, Berlin, Germany, Oct. 5-8, 2004.

[55] R. Choudhury, X. Yang, R. Ramanathan, and N. Vaidya, “On designing mac protocols for wireless network using directional antennas,” *IEEE Transactions on Mobile Computing*, vol. 5, no. 5, pp. 477–491, May 2006.

[56] T. Nandagopal, T.-E. Kim, X. Gao, and V. Bharghavan, “Achieving mac layer fairness in wireless packet networks,” in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom 2000)*, vol. 2, pp. 87–98, Boston, MA, USA, Aug. 6-11, 2000.

[57] V. Loscri and G. Aloi, “Transmission hold-off time mitigation for ieee 802.16 mesh networks: a dynamic approach,” in *Proceedings of the 2008 Wireless Telecommunications Symposium (WTS 2008)*, vol. 2, pp. 31–37, Pomona, CA, April 24-26, 2008.

[58] V. Loscri, “A queue based dynamic approach for the coordinated distributed scheduler of the ieee 802.16,” in *Proceedings of the 2008 IEEE Symposium on Computers and Communications (ISCC 2008)*, vol. 2, pp. 423–428, Marrakech, Morocco, July 6-9, 2008.

[59] H.-M. Lin, W.-E. Chen, and H.-C. Chao, “A dynamic minislot allocation scheme based on ieee 802.16 mesh mode,” in *Proceedings of the Second International Conference on Future Generation Communication and Networking (FGCN ’08)*, vol. 1, pp. 288–293, Sanya, Hainan Island, China, Dec 13-15, 2008.

[60] R. Wang, L. Chen, Y. Li, Z. Liu, and P. Wang, “Conflict improvement methods based on ieee 802.16 mesh networks,” in *Proceedings of the 2009 International Conference on Electronic Computer Technology (ICECT 2009)*, vol. 1, pp. 165–168, Macau, China, Feb. 20-22, 2009.

[61] S.-Y. Wang, C.-C. Lin, and K.-H. Fang, “Improving the data scheduling efficiency of the ieee 802.16(d) mesh network,” in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM 2008)*, vol. 1, pp. 1–5, New Orleans, LA, USA, Nov. 30-Dec. 4 2008.

[62] Q. Xiong, W. Jia, C. Wu, and G. Ye, “Throughput enhancement with bidirectional concurrent transmission in ieee 802.16 mesh networks,” in *Proceedings of the Second International Conference on Communications and Networking in China (CHINACOM 2007)*, vol. 1, pp. 947–951, Shanghai, China, Aug. 22-24, 2007.

Vita

Chih-Che Lin is a Ph.D. candidate of Department of Computer Science, National Chiao Tung University, Taiwan. He received his BS degree in computer science and information engineering from National Chiao Tung University, Taiwan, in 2002, and will receive his Ph.D. degree in computer science from National Chiao Tung University, Taiwan, in July, 2010. He is a core team member of the NCTUns network simulator project. He will work in Industrial Technology Research Institute, Taiwan, in September 2010. His current research interests include wireless networking, wireless mesh networks, vehicular networks, intelligent transportation systems, and network simulation.

Included Publications

- Journal

    1. S.Y. Wang, **C.C. Lin**, H.W. Chu, T.W. Hsu, and K.H. Fang, "Improving the Performances of Distributed Coordinated Scheduling in IEEE 802.16 Mesh Networks," IEEE Transactions on Vehicular Technology, Vol. 57, No. 4, July 2008. (SCI) *(Section 3.1, Section 4.2, and Section 5.2)*

    2. S.Y. Wang, **C.C. Lin**, and C.L. Chou, "Implementing and Evaluating Three Routing Protocols in Dual-Radio-Dual-Mode IEEE 802.11(b) Wireless Mesh Networks," Computer Communication, 31 (2008) 2596 - 2606 (SCI) *(Section 5.4)*

- Conference

    1. S.Y. Wang, **C.C. Lin**, and K.H. Fang, "Improving the Data Scheduling Efficiency of the IEEE 802.16(d) Mesh Network," IEEE Global Communication Conference 2008 (GLOBECOM 2008), November 30 December 4, 2008, New Orleans, LA, USA. *(Section 5.3)*

    2. S.Y. Wang, **C.C. Lin**, K.H. Fang, and T.W. Hsu, "Facilitating the Network Entry and Link Establishment Processes of IEEE 802.16 Mesh Networks," IEEE WCNC 2007 (Wireless Communications and Networking Conference 2007), March 11-15 2007, Hong Kong. *(Section 5.1)*

- Patent

    1. S.Y. Wang, C.D. Tsai, H.H. Weng, **C.C. Lin**, K.H. Fang, and T.W. Hsu, "APPARATUS, METHOD, APPLICATION PROGRAM, AND COMPUTER READABLE MEDIUM THEREOF FOR SIMULTANEOUSLY ESTABLISHING LINKS WITH A PLURALITY OF NODES," patent No. I319669, issued on Jan. 11, 2010. *(Section 5.1)*

- Submitted Article

1. **C.C. Lin**, S.Y. Wang, and T.W. Hsu, "On the Performances of IEEE 802.16(d) Mesh CDS-mode Networks using Single-switched-beam Antennas," submitted to the Springer Wireless Network. *(Section 3.2 and Section 4.3)*

Publication List

- Journal

  1. S.Y. Wang, C.L. Chou, and **C.C. Lin**, "On the Characteristics of Routing Paths and the Performance of Routing Protocols in Vehicle-Formed Mobile Ad Hoc Networks on Highways," Wireless Communications and Mobile Computing, Wiley, (accepted and to appear, already published online on March 24, 2009) (SCI)
  2. S.Y. Wang, C.L. Chou, and **C.C. Lin**, "Increasing Wide-area Download Throughputs on the Roads by Trunking Multiple Cellular Channels over A Vehicular Ad Hoc Network," Computer Communication, Vol. 32, Issue 2, p.p. 268-280, 2009 (SCI)
  3. S.Y. Wang, **C.C. Lin**, and C.L. Chou, "Implementing and Evaluating Three Routing Protocols in Dual-Radio-Dual-Mode IEEE 802.11(b) Wireless Mesh Networks," Computer Communication, 31 (2008) 2596 - 2606 (SCI)
  4. S.Y. Wang, **C.C. Lin**, H.W. Chu, T.W. Hsu, and K.H. Fang, "Improving the Performances of Distributed Coordinated Scheduling in IEEE 802.16 Mesh Networks," IEEE Transactions on Vehicular Technology, Vol. 57, No. 4, July 2008. (SCI)
  5. S.Y. Wang, C.L. Chou, **C.C. Lin**, "The Design and Implementation of the NCTUns Network Simulation Engine," Simulation Modeling Practice and Theory. 15 (2007), pp. 57-81. (SCI)
  6. S.Y. Wang, C.L. Chou, C.H. Huang, C.C. Hwang, Z.M. Yang, C.C. Chiou, and **C.C. Lin**, "The Design and Implementation of the NCTUns 1.0 Network Simulator," Computer Networks, Vol. 42, Issue 2, June 2003, pp. 175-197. (SCI)

- Book Chapter

  1. S.Y. Wang, **C.C. Lin**, and C.C. Huang, "NCTUns Tool for Evaluating the Performances of Real-life P2P Applications," a chapter of the "Peer-to-Peer

Networks and Internet Policies" book, Nova Science Publisher, ISBN: 978-1-60876-287-3.

- Conference

1. S.Y. Wang, **C.C. Lin**, P.S. Koo, Y.M. Huang, and J.S. Yang, "NCTUns Emulation Testbed for Evaluating Real-life Applications over WiMAX Networks," IEEE PIMRC 2010 (The 21th IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications), September 26 29, 2010, Istanbul, Turkey.
2. S.Y. Wang, **C.C. Lin**, and C.C. Huang, "The Effects of Underlying Physical Network Topologies on Peer-to-peer Application Performances," IEEE PIMRC 2010 (The 21th IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications), September 26 29, 2010, Istanbul, Turkey.
3. S.Y. Wang, **C.C. Lin**, K.C. Liu, and W.J. Hong, "On Multi-hop Forwarding over WBSS-based IEEE 802.11(p)/1609 Networks," the 20th IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC 2009), September 13 16, 2009, Tokyo, Japan.
4. S.Y. Wang, **C.C. Lin**, and K.H. Fang, "Improving the Data Scheduling Efficiency of the IEEE 802.16(d) Mesh Network," IEEE Global Communication Conference 2008 (GLOBECOM 2008), November 30 December 4, 2008, New Orleans, LA, USA.
5. S.Y. Wang and **C.C. Lin**, "NCTUns 5.0: A Network Simulator for IEEE 802.11(p) and 1609 Wireless Vehicular Network Researches," 2nd IEEE International Symposium on Wireless Vehicular Communications (WiVeC 2008), September 21-22, 2008, Calgary Marriott, Canada.
6. S.Y. Wang, H.L. Chao, K.C. Liu, T.W. He, **C.C. Lin** and C.L. Chou, "Evaluating and Improving the TCP/UDP Performances of IEEE 802.11(p)/1609 Networks," IEEE ISCC 2008 (IEEE Symposium on Computers and Communications 2008), July 6-9, 2008, Marrakech, Morocco (Awarded as the best paper).
7. S.Y. Wang, Y.W. Chen, **C.C. Lin**, W.J. Hong, and T.W. He, "A Vehicle

Collision Warning System Employing Vehicle-to-Infrastructure Communications," IEEE WCNC 2008 (Wireless Communications and Networking Conference 2008), March 31-April 3, 2008, Las Vegas, USA.

8. S.Y. Wang, **C.C. Lin**, K.H. Fang, and T.W. Hsu, "Facilitating the Network Entry and Link Establishment Processes of IEEE 802.16 Mesh Networks," IEEE WCNC 2007 (Wireless Communications and Networking Conference 2007), March 11-15 2007, Hong Kong.

9. S.Y. Wang, **C.C. Lin**, Y.W. Hwang, K.C. Tao, and C.L. Chou, "A Practical Routing Protocol for Vehicle-Formed Mobile Ad Hoc Networks on the Roads," IEEE ITSC 2005 (International Conference on Intelligent Transportation Systems), September 13-16 2005, Vienna, Austria.

- Patent

  1. S.Y. Wang, C.D. Tsai, H.H. Weng, **C.C. Lin**, K.H. Fang, and T.W. Hsu, "APPARATUS, METHOD, APPLICATION PROGRAM, AND COMPUTER READABLE MEDIUM THEREOF FOR SIMULTANEOUSLY ESTABLISHING LINKS WITH A PLURALITY OF NODES," patent No. I319669, issued on Jan. 11, 2010.

- Submitted Article

  1. **C.C. Lin**, S.Y. Wang, and T.W. Hsu, "On the Performances of IEEE 802.16(d) Mesh CDS-mode Networks using Single-switched-beam Antennas," submitted to the Springer Wireless Network.