在 NCTUns 平台上模擬 WiMAX 網狀網路
Simulating WiMAX Mesh Networks over the NCTUns Network Simulator

研究生：朱漢威　　Student：Han-Wei Chu

指導教授：王協源　　Advisor：Shie-Yuan Wang

國立交通大學

資訊科學與工程研究所

碩士論文

A Thesis

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

Master

in

Computer Science

June 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年六月

在NCTUns平台上模擬WiMAX網狀網路

Simulating WiMAX Mesh Networks over the NCTUns Network Simulator

學生：朱漢威　　　　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碩士班

摘要

IEEE 802.16 (也稱爲WiMAX)是一個前景看好的固定式寬頻無線接取系統的標準，在這個標準之中，制定了兩種運作模式。第一種是點對多點模式，用以取代傳統固接式有線的「最後一哩」技術；第二種是網狀模式，適用於下一世代的無線都會型區域網路。與傳統的IEEE 802.11網狀網路相比，IEEE 802.16網狀網路提供了更高的可用頻寬及更大的涵蓋範圍。在WiMAX網路上有諸多研究議題，如服務品質(Quality of Service)、空間再利用(spatial reuse)及網路效能等。然而，目前仍然沒有一個公開的網路模擬器可以支援WiMAX網路的模擬。在這篇論文中，我們在NCTUns網路模擬器上設計及實作了一個WiMAX網狀網路的模擬系統，其所提供的功能經過了驗證。我們評估了在不同的模擬環境下的網路效能，並且討論系統的可擴充性及未來的發展方向。

關鍵字：網路模擬器，網狀網路，無線都會型區域網路，IEEE 802.16，WiMAX。

Simulating WiMAX Mesh Networks over the NCTUns Network Simulator

Student: Han-Wei Chu   Advisors: Prof. Shie-Yuan Wang

Institute of Computer Science and Engineering
National Chiao Tung University

ABSTRACT

The IEEE 802.16d standard (also known as WiMAX) is a promising technology for future fixed broadband wireless access (FBWA) systems. There are two operation modes defined in this standard. First, the point-to-multipoint (PMP) mode aims to replace the traditional wired last-mile solutions. Second, the Mesh mode is designed for the next-generation wireless metropolitan area networks (Wireless-MANs). Compared with the traditional 802.11-based mesh network, the 802.16 Mesh network provides higher throughput and larger coverage.

Many critical issues of WiMAX networks such as quality of service (QoS), spatial reuse, and network performance are being broadly studied by researchers. However, there is still no general-purposed and publicly available network simulators for WiMAX networks.

In this thesis, we design and implement a WiMAX Mesh simulation system over the NCTUns network simulator. Functionalities of our implementation are validated. We evaluate the performance of WiMAX Mesh networks under different conditions. Scalability issues are also discussed. Finally, possible future developments are proposed.

Keywords: network simulator, mesh networks, wireless metropolitan area networks, IEEE 802.16, WiMAX.

誌謝辭

首先，要感謝我的指導教授王協源老師、周智良學長及林志哲學長，研究所的這兩年，從你們身上學到許多寶貴的經驗與知識，相信對於我未來就業或者繼續求學都有很大的助益。

感謝蔡宗易學長、周智良學長及林志哲學長，在百忙之中抽空，陪我修改投影片及進行口試預演。感謝甘苦同享的室友們，希望以後還能有機會聚在一塊。

感謝盈詮，這兩年的相互支持與陪伴，讓我們看到彼此更多的優點。感謝阿媽、舅舅、阿姨、鄺叔叔一家人、汪阿姨一家人及姑姑一家人，謝謝你們的栽培與鼓勵。最後要謝謝親愛的老媽，妳的愛與關懷是我在求學路上最大的動力。

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The IEEE 802.16d standard [1], recently standardized by the IEEE 802.16 working group, is a promising technology for future FBWA systems. The standard defines two operation modes, namely the PMP mode and the Mesh mode. The PMP mode is a novel last-mile technology to replace traditional wired solutions. The downlink, from the base station (BS) to subscriber stations (SSs), operates on a PMP basis, while the uplink is shared by all SSs. On the other hand, the Mesh mode is designed for constructing the infrastructure of Wireless-MANs. In this mode, traffic can be routed through other SSs and can occur directly between SSs.

WiMAX networks have many critical issues such as QoS, spatial reuse, and network performance. Researchers focusing on these issues usually develop their own simulation programs either to evaluate their proposed schemes or to validate their analytical methods [3] [6] [8] [11]. Unfortunately, these simulation programs are often *special-purposed or not publicly available*. Therefore, a high-quality simulator capable of simulating WiMAX networks is desired. It is valuable to researchers who develop, test or improve the mechanisms specified in the standard. Vendors can also carry out important experiments by simulations before the actual hardware deployment takes place.

Motivated by the above observation, in this thesis we design and implement a WiMAX Mesh simulation system over the NCTUns network simulator [10]. Functionalities such as coordinated distributed scheduling, control message exchanging, and the network entry process are implemented and validated. When coordinated distributed scheduling is used, peers in the mesh network exchange their schedule information via MSH-DSCH messages. The transmission timing of such messages is determined by the distributed election-based transmission timing (EBTT) mechanism. The mechanism is fair and robust. Also, it ensures that the exchange is free from collision. By running a simulation case consisting of 25 nodes, we show that the implementation meets the standard specification.

Our contributions are threefold. First, this is the first public WiMAX simulation system and is developed over a general-purposed network simulator. Second, our WiMAX simulation system is validated by many simulation cases under different conditions. Third, we present the detailed design and implementation of this system. Other researchers can improve the system based on our work.

The rest of this thesis is organized as follows. In Chapter 2, the background on the IEEE 802.16 Mesh mode and the NCTUns network simulator are presented. We describe the detailed design and implementation in Chapter 3, then validate the implementation in Chapter 4. Chapter 5 presents experiments to evaluate the performance of WiMAX Mesh networks and Chapter 6 discusses scalability issues. Finally, we propose possible extensions to our simulation modules in Chapter 7 and conclude in Chapter 8.

Chapter 2

Background

The IEEE 802.16d standard defines the specification of the air interface for FBWA systems. Two operation modes are specified in the standard. Firstly, the PMP mode is designed to replace the traditional wired last-mile. When operating in the 10-66 GHz licensed bands, its line-of-sight (LOS) applications offer data rates greater than 120 Mb/s. Secondly, the Mesh mode is defined to support multihop wireless communication. Compared with the traditional 802.11-based mesh network, the 802.16 Mesh network provides higher throughput and larger coverage, which makes it a promising technology for the next-generation Wireless-MANs.

Fig. 2.1 illustrates the reference model proposed in the standard. The MAC layer consists of three sublayers, namely the service-specific convergence sublayer (CS), the MAC common part sublayer (MAC CPS), and the security sublayer. The PHY

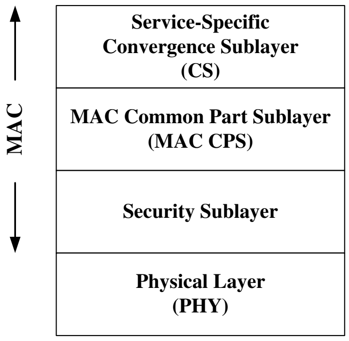

MAC
Service-Specific Convergence Sublayer (CS)
MAC Common Part Sublayer (MAC CPS)
Security Sublayer
Physical Layer (PHY)

Figure 2.1: The IEEE Std 802.16 protocol layering

layer adopts technologies such as single-carrier and orthogonal frequency division multiplexing (OFDM), supporting both LOS and non-LOS applications.

In this chapter, we describe the MAC layer and the PHY layer operations of the IEEE 802.16 Mesh mode, especially the MAC CPS. An introduction to the NCTUns network simulator and its module-based platform is also presented.

2.1 IEEE 802.16 Mesh Mode Operations

2.1.1 Service-Specific Convergence Sublayer

The standard specifies two CS specifications: the asynchronous transfer mode (ATM) CS and the packet CS. The major function performed by this sublayer is classifying data from the higher layer and associating them to appropriate MAC connections. Optionally, the repetitive portion of the payload headers of the higher layer can be suppressed by the sending CS and restored by the receiving CS.

The packet CS supports transport for packet-based protocols such as Internet Protocol (IP), Point-to-Point Protocol (PPP), and Ethernet. *In this thesis, we only consider the IP CS.*

2.1.2 MAC Common Part Sublayer

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

- Mesh BS and Mesh SS

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

- New Node, Candidate Node, and Sponsoring Node

  In the network entry process, a New Node becomes a Candidate Node after selecting some regular node as its sponsor. The Sponsoring Node assists the Candidate Node in entering the WiMAX Mesh network.

- Registration Node

  A Registration Node is a Mesh node that assigns Node IDs to newcomers of the WiMAX Mesh network.

- Connection

  In the PMP mode, a connection is a unidirectional mapping between two MAC peers. It is associated with a set of QoS parameters and identified by a 16-bit connection identifier (CID). The CID is carried in the MAC header and indicates the target destination.

- Link

  In the Mesh mode, a link, instead of a connection, is established between two nodes. QoS is provisioned over links on a *message-by-message* basis. A link is identified by an 8-bit link identifier (Link ID) and used to construct the CID, as shown in Fig. 2.2.

- Node ID

  After a node is authorized to enter the network, it shall receive a 16-bit node identifier (Node ID).Thereafter, the Node ID is transferred together with every outgoing MAC protocol data unit (PDU). This unique identifier is used to indicate the identity of the transmitting node.

- Neighbor, neighborhood, and extended neighborhood

  The stations with which a node has direct links are called the node's neighbors. Neighbors of a node form a neighborhood and are considered to be one-hop away from the node. An extended neighborhood contains, additionally, all the neighbors of the nodes in the neighborhood.

Before detailed operations of the MAC CPS are elaborated, the format of the MAC PDU and the frame structure are presented first. As shown in Fig. 2.3, the MAC PDU comprises a 6-byte generic MAC header, zero or more subheaders (if

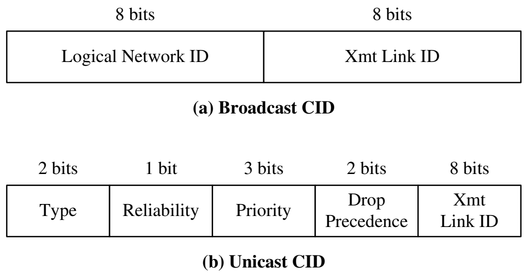


Figure 2.2: Mesh CID construction: (a) broadcast CID, and (b) unicast CID

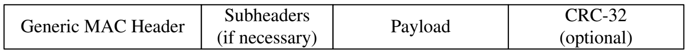


Figure 2.3: The MAC PDU format

necessary), a payload, and an optional 32-bit cyclic redundancy check (CRC). The generic MAC header indicates whether subheaders are presented in the message and whether the CRC is appended. Possible subheaders are shown in Table 2.1.

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

The payload carries either data from the higher layer or MAC management messages. Table 2.2 lists important management messages used in the Mesh mode.

Table 2.1: MAC subheaders

Subheader type	**Carried information**
Fragmentation	The fragmentation state of the payload
Grant	Bandwidth management needs
Packing	The fragmentation state of the payload
Mesh	Xmt Node ID
FAST-FEEDBACK	PHY-specific information

Table 2.2: MAC management messages used in the Mesh mode

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

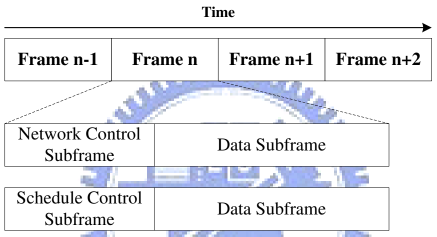


Figure 2.4: The Mesh frame structure

Frame Structure

The Mesh frame structure is illustrated in Fig. 2.4 and Fig. 2.5. Only time division duplex (TDD) is supported in the Mesh mode. A Mesh frame consists of a control subframe and a data subframe. There are two kinds of control subframes, namely the network control subframe and the schedule control subframe. The former is used for network configuration, node entry, and synchronization. The latter facilitates the exchange of coordinated schedule information.

The MSH-NCFG message contains a Network Descriptor that dictates the detailed frame format. The length of the control subframe is a fixed value of length MSH-CTRL-LEN transmit opportunities (TxOpps). Each TxOpp consists of 7 OFDM symbols. Scheduling-Frames defines the period of frames with a network

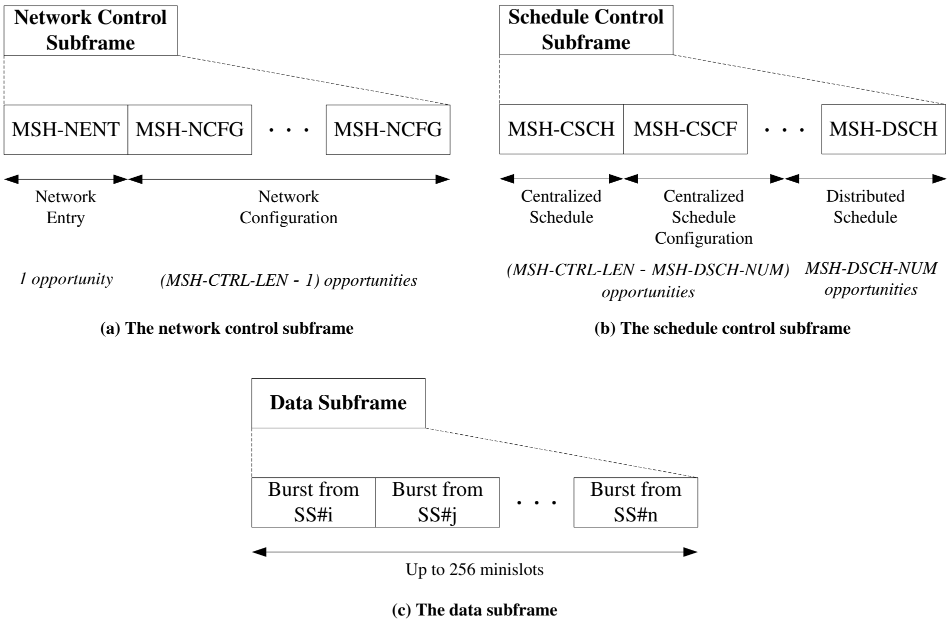


Figure 2.5: Mesh subframes in detail

control subframe. All other frames contain a schedule control subframe. In the network control subframe, there is one TxOpp for the network entry process, followed by (MSH-CTRL-LEN - 1) TxOpps for the network configuration. MSH-DSCH-NUM defines the number of MSH-DSCH TxOpps per schedule control subframe. During the schedule control subframe, the first (MSH-CTRL-LEN - MSH-DSCH-LEN) TxOpps are reserved for centralized scheduling. The remainder is allocated to distributed scheduling.

Scheduled data transmissions and uncoordinated scheduling packets take place in the data subframe, which is divided into (up to) 256 minislots. A scheduled allocation consists of one or more minislots.

Network Entry Process

A new node shall perform the network entry process before it can start a scheduled transmission. In this process, the New Node first listens to the ongoing transmissions in the air, searching for MSH-NCFG messages to synchronize with the network. In the meantime, the New Node shall build a physical neighbor list according to the information carried in the MSH-NCFG message. When enough information is acquired, the New Node selects a potential Sponsoring Node from the list, and itself becomes a Candidate Node. By exchanging MSH-NENT and MSH-NCFG messages, the Candidate Node and the Sponsoring Node establish a temporary Sponsor Channel. From that moment on, activities between the two peers are over the Sponsor Channel.

After basic capabilities (such as ARQ support and CRC support) are negotiated and authorization is performed, the registration procedure in Fig. 2.6 is followed. The Candidate Node transmits a REG-REQ message to register with the Registration Node. Upon receiving the REG-REQ message the Sponsor shall tunnel the message by prepending a tunnel subheader, a UDP header and a IP header. This tunneled message is sent to the Registration Node, which can optionally be co-located with the Mesh BS. Upon receiving the REG-RSP message from the Registration Node, the Sponsor shall extract the message and forward it to the Candidate Node, which then retrieves its Node ID from the message. The Candidate Node continues to establish IP connectivity via DHCP, retrieve the current system time via the protocol defined in IETF RFC 868, and download a file containing operational parameters via TFTP. Finally, the Sponsor Channel is closed and the Candidate Node becomes a regular node in the network. Links between this newcomer and its neighbors can be later established by exchanging MSH-NCFG messages.

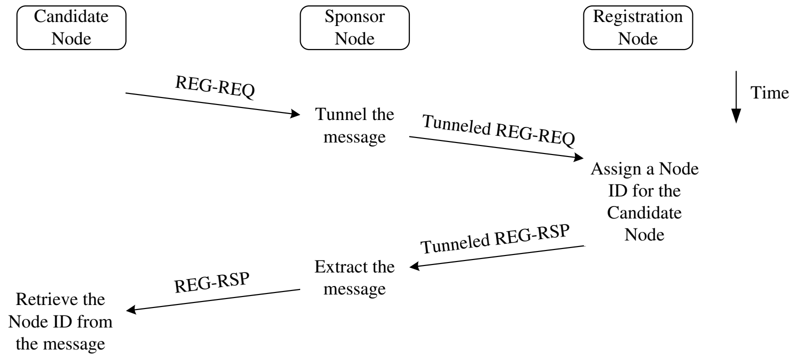


Figure 2.6: The registration procedure

Network Synchronization

Regular nodes in the network periodically broadcast MSH-NCFG messages to exchange network configuration information with their neighbors. The transmission timing (i.e. the TxOpp) of the MSH-NCFG message is determined by a distributed EBTT mechanism, which works without explicit negotiation and is completely distributed, fair, and robust. This mechanism ensures the resulting transmission is collision-free within the extended neighborhood (2-hop neighborhood) of each node.

To reduce the signalling overhead, a node does not broadcast its exact schedule. Instead, only a 5-bit **Next Xmt Mx** and a 3-bit **Xmt Holdoff Exponent** are advertised in the MSH-NCFG message. The interval of the next transmit time (**Next Xmt Time**) of the node (i.e. the sender of the MSH-NCFG message) can be computed as follows:

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

The **Xmt Holdoff Time** is the number of MSH-NCFG TxOpps within which this node is *not eligible* to transmit MSH-NCFG packets after the **Next Xmt Time**.

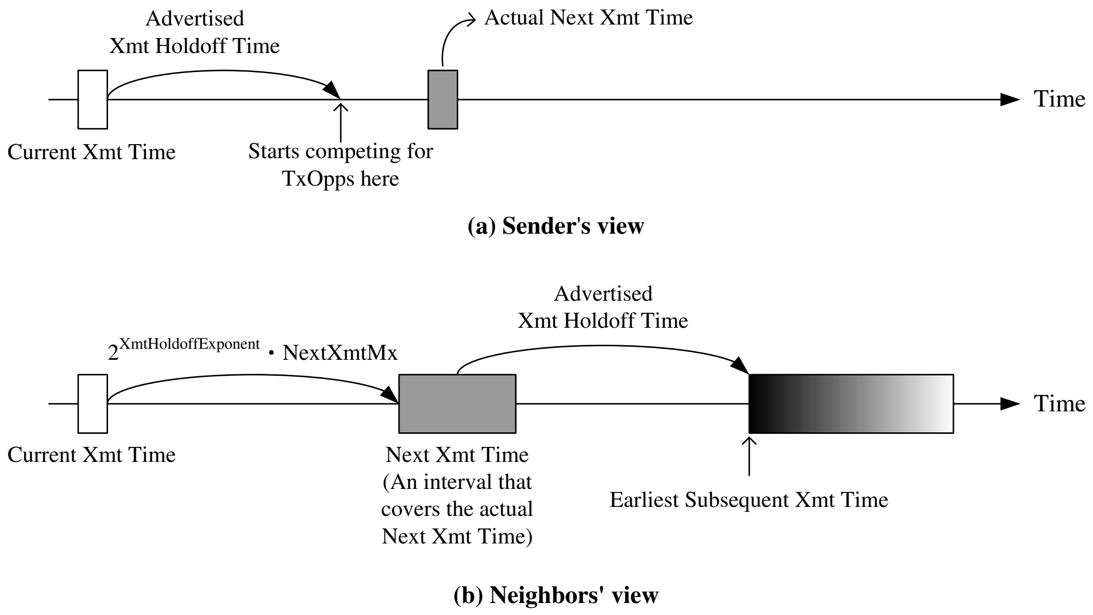
$$XmtHoldOffTime = 2^{XmtHoldoffExponent+4}$$

In other words, the node cannot transmit any MSH-NCFG messages in this period. Thus, we can obtain the node's **Earliest Subsequent Xmt Time** by

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

To make it clearer, to the sender itself, the **Next Xmt Time** is pointing to an exact TxOpp. To its neighbors, however, the **Next Xmt Time** of the sender is an interval that covers the actual **Next Xmt Time** of the sender. The timing is illustrated in Fig. 2.7.

A node broadcasts not only its own **Next Xmt Mx** and **Xmt Holdoff Exponent** but also these two values of all its one-hop neighbors. By this way, every regular node possesses the scheduling information within its extended neighborhood. When the node is about to transmit the MSH-NCFG message, it shall schedule its next



(a) Sender's view

(b) Neighbors' view

Figure 2.7: Current Xmt Time, Next Xmt Time, and Earliest Subsequent Xmt Time

MSH-NCFG transmission using these scheduling information as shown in Fig. 2.8. By definition, the node can only compete for TxOpps later than the **Current Xmt Time** plus the node's advertised **Xmt Holdoff Time** (Fig. 2.7). That's where the **Temp Xmt Time** starts from. Neighbors that are eligible to compete with the sender for the **Temp Xmt Time** are those who meet one or more of the following conditions:

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

A Mesh Election for the **Temp Xmt Time** is held among the sender and the eligible competing nodes. A pseudorandom mixing number $f(\mathit{Node\ ID}, \mathit{Temp\ Xmt\ Time})$ is computed for each of the nodes involved in the election. If $f(\mathit{Sender's\ Node\ ID}, \mathit{Temp\ Xmt\ Time})$ is the greatest among all numbers, the sender set its **Next Xmt Time** equal to

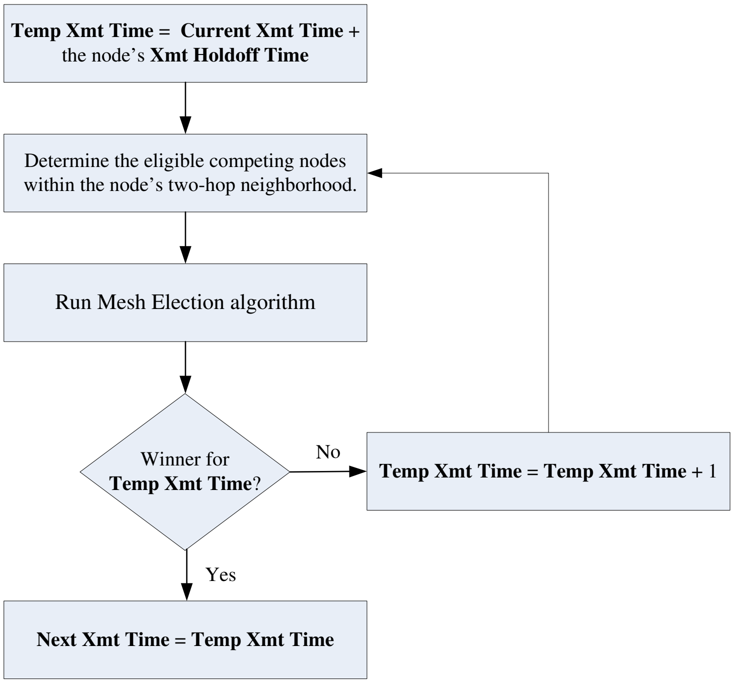

Temp Xmt Time = Current Xmt Time + the node's Xmt Holdoff Time
Determine the eligible competing nodes within the node's two-hop neighborhood.
Run Mesh Election algorithm
Winner for Temp Xmt Time?
No
Temp Xmt Time = Temp Xmt Time + 1
Yes
Next Xmt Time = Temp Xmt Time

Figure 2.8: Scheduling next MSH-NCFG transmission

Temp Xmt Time. Otherwise, the **Temp Xmt Time** is advanced and the algorithm is repeated. Note that the fairness is ensured by the way the pseudorandom mixing numbers are computed. The seeds (**Temp Xmt Time**) are different for every TxOpp.

After the **Next Xmt Time** is decided, it is converted into the corresponding **Next Xmt Mx**. The **Next Xmt Mx** and the *fixed* **Xmt Holdoff Exponent** are then added to the outgoing MSH-NCFG message.

The MAC CPS coordinates the access to the shared wireless transmission media. Network resources may be allocated in a centralized fashion, a distributed fashion, or a combination of both. These mechanisms also determine the way data PDUs are routed in the network.

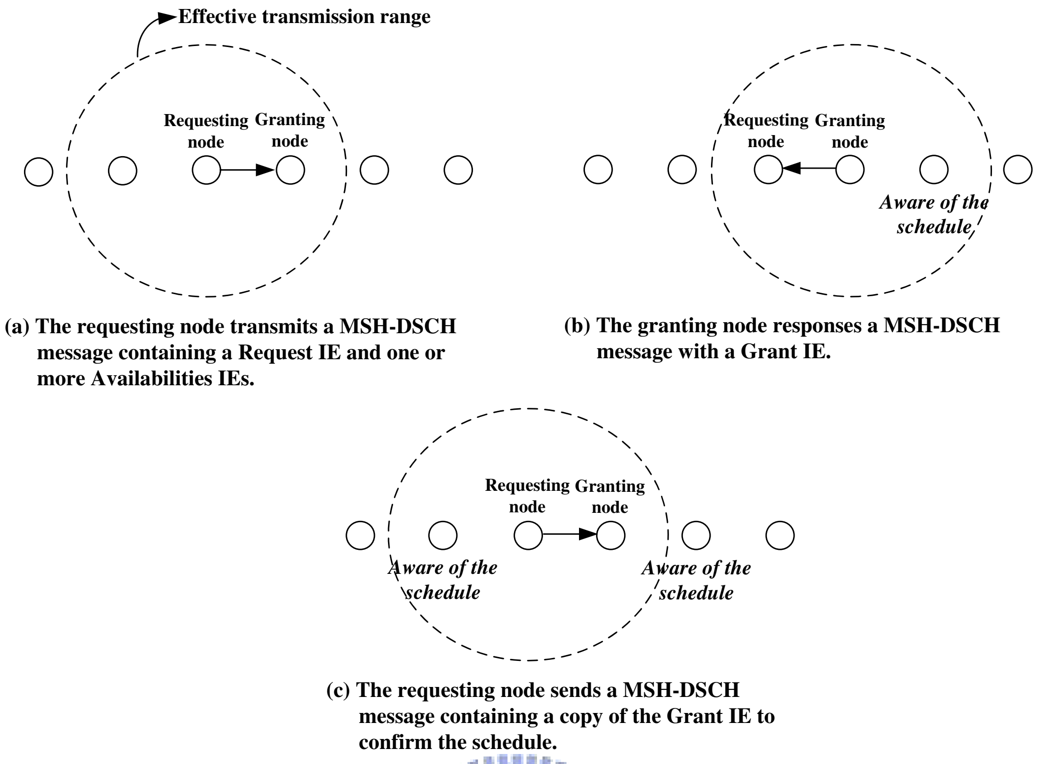
When centralized scheduling is used, the Mesh BS acts as a central coordinator to allocate network resources. It builds a scheduling tree made of nodes within a certain hop range and gathers resource requests in a bottom-up way. After determining the amount of granted resources for each link in the scheduling tree, the Mesh BS broadcasts the schedule information, which are rebroadcasted by other nodes if needed. In contrast, when the network adopts distributed scheduling, all nodes including the Mesh BS are peers. Schedules are established in a distributed way. A three-way handshake procedure ensures the resulting transmissions do not cause collisions within the extended neighborhood. In the following we describe distributed scheduling in more detail.

Distributed Scheduling

In coordinated distributed scheduling, all nodes including the Mesh BS transmit a MSH-DSCH message periodically in the control subframe to announce their schedules. *The transmission timing is determined by the same algorithm used for MSH-NCFG messages.* Therefore, the resulting transmissions are collision-free. In the uncoordinated case, MSH-DSCH messages are exchanged in the data subframe. Collisions may occur if two or more nearby stations send their messages at the same time.

There are four kinds of information elements (IEs) that can be included in the MSH-DSCH message. The Scheduling IE carries the coordinated distributed scheduling information: **Next Xmt Mx** and **Xmt Holdoff Exponent**. Each node reports these two parameters of its own and all its one-hop neighbors. The Request IE is used to convey resource requests on a link, with the demand expressed in terms of minislots. The Availabilities IE indicates free minislot ranges of the requesting node. The granting node uses the Grant IE to indicate the range of granted minislots selected from the free minislots reported by the requesting node. When sent by the requesting node, the Grant IE acts as a grant confirmation.

In both coordinated and uncoordinated cases, schedules are established between two nodes using a three-way handshake procedure. The requesting node first transmits a MSH-DSCH message containing a Request IE and one or more Availabilities IEs. Upon reception of this message, the granting node responds a MSH-DSCH message with a Grant IE, indicating the actual grant. Neighbors of the granting node



(a) The requesting node transmits a MSH-DSCH message containing a Request IE and one or more Availabilities IEs.

(b) The granting node responses a MSH-DSCH message with a Grant IE.

(c) The requesting node sends a MSH-DSCH message containing a copy of the Grant IE to confirm the schedule.

Figure 2.9: The three-way handshake procedure to establish a schedule

(except the requesting node) shall assume the schedule takes places as granted. The requesting node then sends a MSH-DSCH message containing a copy of the Grant IE to confirm the schedule to the third party (i.e. its neighbors except the granting node). This three-way handshake procedure ensures data transmissions in the established schedule are collision-free. Fig. 2.9 illustrates this procedure.

2.1.3 Security Sublayer

The security sublayer provides subscribers with privacy across the fixed broadband wireless network by encrypting connections between stations. It also provides operators with strong protection from theft of service. The capability to support 802.16 security is negotiated during the network entry process.

2.1.4 Physical Layer

The physical layer for the Mesh mode is operating in the licensed bands below 11GHz and based on the OFDM technology. OFDM with a 256 point transform is used to overcome delay spread, multipath, and inter-symbol interference (ISI) in this physical environment.

To better utilize the channel, a typical channel coding scheme is included in the standard, as shown in Fig. 2.10. The randomizer first scrambles the bit stream to avoid long runs of zeros or ones. The encoding is performed by passing the scrambled data blocks through the Reed-Solomon (RS) encoder and then passing the RS-encoded blocks through the convolutional code (CC) encoder. The RS code is a shortened and punctured code derived from a systematic $RS(N = 255, K = 239, T = 8)$ code using $GF(2^8)$. The CC is a punctured code derived from the basic CC 1/2. Various correcting capabilities can thus be realized by this concatenated coding scheme. The coded block is further interleaved to avoid long runs of bit errors. Finally, bits are entered serially to the constellation mapper.

Mandatory PHY modes are listed in Table 2.3 in the order of decreasing robustness (or increasing efficiency).

2.2 The NCTUns Network Simulator

The NCTUns network simulator is a high-fidelity and extensible network simulator. By using a novel kernel re-entering simulation methodology, a real-life UNIX kernel's protocol stack can be directly used to generate simulation results. Also, real-life UNIX application programs can run on top of the simulation environment without any modification.

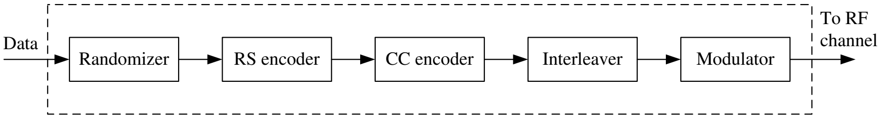


Figure 2.10: Channel coding scheme

Table 2.3: Mandatory PHY modes

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

The simulation engine of the NCTUns network simulator uses the discrete-event simulation method to advance its virtual clock. Therefore, its performance depends on the number of events it needs to process. The more events it needs to process, the slower its simulation speed will be.

It adopts an open-system architecture and provides a module-based platform. Fig. 2.11 is an example that depicts a network topology consisting of three nodes and the organization of each node. In the module-based platform, a module skeleton is provided as shown in Fig. 2.12. Based on the skeleton, researcher can easily develop their own protocol modules and integrate them into the simulator.

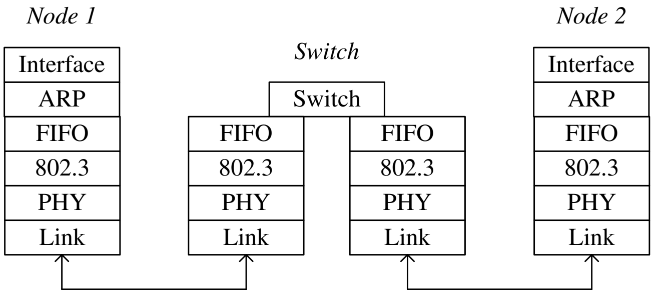
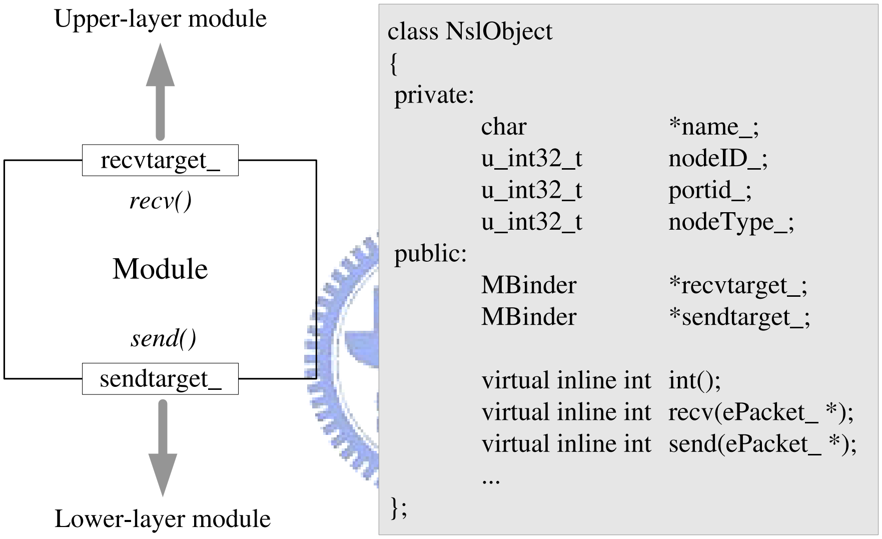


Figure 2.11: The module-based platform provided by the NCTUns network simulator



```
class NslObject
{
 private:
            char                *name_;
            u_int32_t           nodeID_;
            u_int32_t           portid_;
            u_int32_t           nodeType_;
 public:
            MBinder             *recvtarget_;
            MBinder             *sendtarget_;

            virtual inline int  int();
            virtual inline int  recv(ePacket_ *);
            virtual inline int  send(ePacket_ *);
            ...
};
```

Figure 2.12: The module skeleton provided by the NCTUns network simulator

Chapter 3

Design and Implementation

This chapter presents our WiMAX Mesh reference implementation over the NCTUns network simulator. We first describe a high level architecture of the entire system and then the detailed design and implementation of our protocol modules.

3.1 High Level Architecture

3.1.1 Network Scenario

Infrastructure meshing, *client meshing*, and *hybrid meshing* are three common approaches to deploy wireless mesh networks (WMNs) [2]. In infrastructure WMNs, mesh routers and gateways constitute a high-speed wireless mesh backbone for mesh clients, providing connectivity to other networks such as the Internet and cellular networks. On the contrary, mesh clients themselves constitute the network in client WMNs. Hybrid WMNs, combining these two approaches, provide the best flexibility to allow mesh clients either to access the network through mesh routers or to communicate with each other in a peer-to-peer fashion.

A typical scenario of hybrid WiMAX Mesh networks is shown in Fig. 3.1. The Mesh BS Gateway has a direct connection to the Internet and constitutes a wireless mesh backbone together with Mesh SS Gateways and Forwarders. Mesh SS Clients connected to the backbone can also perform direct meshing with each other.

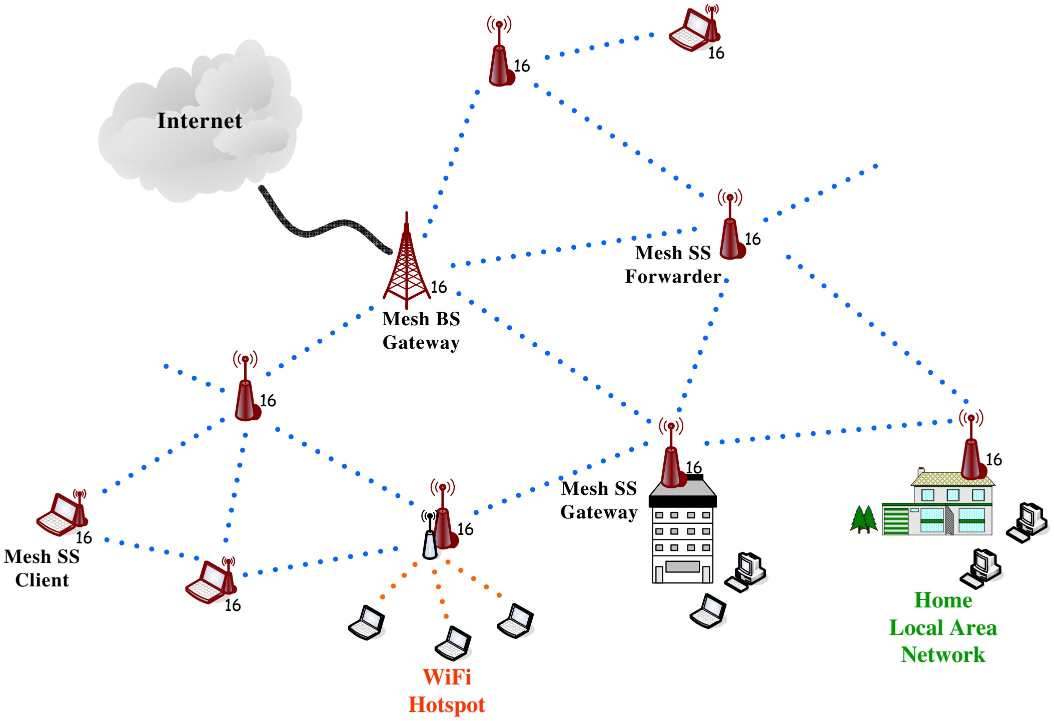


Figure 3.1: Mesh network scenario

The differences between the Mesh SS Gateway and the Mesh SS Forwarder are as follows: A Mesh SS Forwarder performs functions such as multihop routing and self-configuration. A Mesh SS Gateway, on the other hand, is a Mesh SS Forwarder that additionally performs gateway functions and connects private or public networks to the wireless mesh backbone.

In our reference implementation of WiMAX Mesh networks, we support the four types of mesh nodes mentioned above. Using the fully-integrated GUI environment of the NCTUns network simulator, researchers can easily design a network topology similar to Fig. 3.1.

3.1.2 Routing Scheme

A mesh network with layer-2 routing leads to a layer-2 network in which mobility management can be easily achieved. Nevertheless, in a big layer-2 network, common protocols such as DHCP, ARP and RARP that require layer-2 broadcast functional-

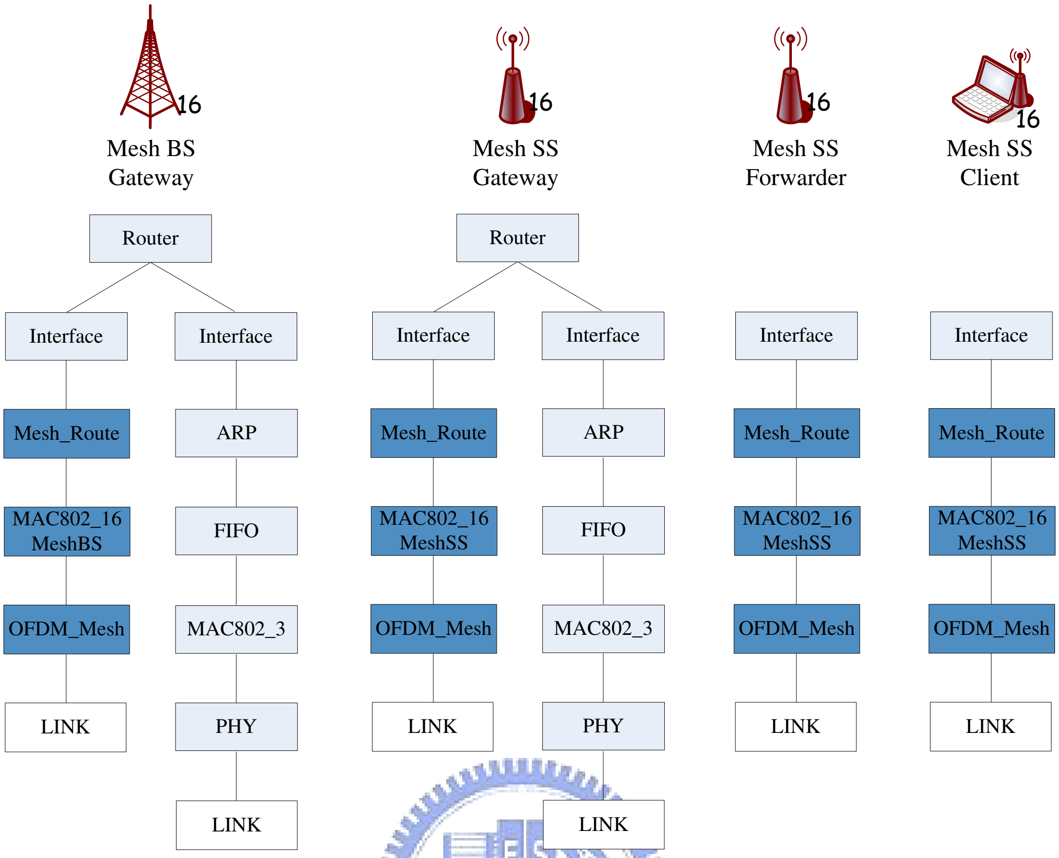


Figure 3.2: Four types of mesh nodes supported and their corresponding protocol stacks

ity can result in bandwidth wastage [9]. Therefore, in our reference implementation of WiMAX Mesh networks, we adopt layer-3 routing which does not suffer from this drawback. Mobility management is beyond the scope of this thesis. However, it can be achieved via a Mobile IP based solution.

In our current implementation, a *fixed* layer-3 routing scheme is used. This can be later replaced by other layer-3 routing protocols such as AODV, OSPF and RIP.

3.1.3 Protocol Stacks

The four types of mesh nodes supported and their corresponding protocol stacks are shown in Fig. 3.2. Protocol modules colored with deep blue are WiMAX-related modules. Others are existing and well-tested modules provided by the original NC-

TUns network simulator. In the following we briefly describe some of these modules.

- *`Mesh_Route`*

  Before a simulation starts, a fixed routing table is predetermined for the target network topology. During the simulation, ***`Mesh_Route`*** looks up this table to find the next-hop Node ID of a received IP packet. This process effectively associates IP packets to appropriate MAC connections (or links). In other words, functions of the IP CS are included in this module.

- *`MAC802_16_MeshSS`*

  This module performs important functions including the network entry process, the Mesh Election algorithm, and distributed scheduling.

- *`MAC802_16_MeshBS`*

  This module performs all functions of ***`MAC802_16_MeshSS`*** and is responsible for assigning Node IDs to newcomers of the WiMAX Mesh network.

- *`OFDM_Mesh`*

  This module performs channel coding and simulates a path loss model for the underlying wireless channel.

- *`Router`*

  This is not a real protocol module. Instead, it represents the layer-3 routing functions provided by the operating system which are only used in *wired networks.*

- *`Interface`*

  This can be view as a network interface owned by a router or a host.

3.1.4 Packet Processing flows

Here we give two examples of packet processing flows. Fig. 3.3 shows a non-forwarding example. A downstream IP packet coming from the Internet first enters

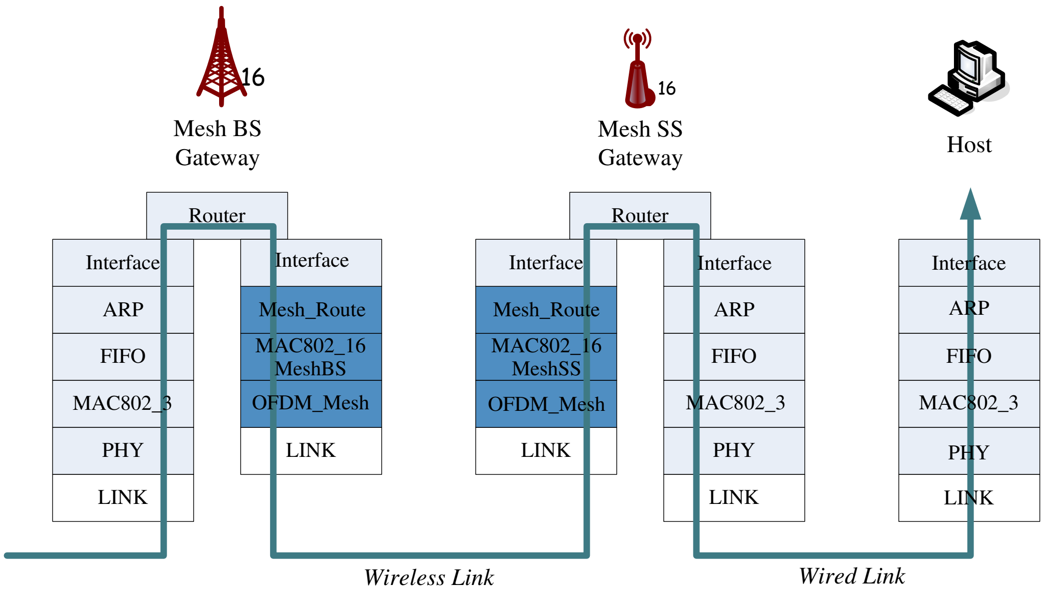


Figure 3.3: A non-forwarding example of packet processing flows

the wired protocol stack of the Mesh BS Gateway. The layer-3 routing mechanism provided by the operating system directs this packet to the WiMAX protocol stack. *Mesh_Route* then finds the next-hop of the packet and passes the packet together with this information to *MAC802_16_MeshBS*. At a scheduled minislot range, the packet is delivered to *OFDM_Mesh*. After channel coding is applied, the packet is transmitted to the air.

Upon reception of the packet, *OFDM_Mesh* of the Mesh SS Gateway applies channel decoding to the packet and delivers the packet to *MAC802_16_MeshSS*. After CRC and other tests are done in *MAC802_16_MeshSS*, the packet is passed to *Mesh_Route*. *Mesh_Route* finds that the destination of the packet is located behind the Mesh SS Gateway, so it further passes the packet to the upper layer. The remaining processing is the same as the original packet processing flows of wired networks in the NCTUns network simulator.

Fig. 3.4 illustrates a forwarding example. It can be seen that in the Mesh SS Forwarder the received packet is redirected in *Mesh_Route*. Note that the Mesh BS Gateway and the Mesh SS Gateway also perform the forwarding function. In other words, they also forward packets in the multihop wireless network.

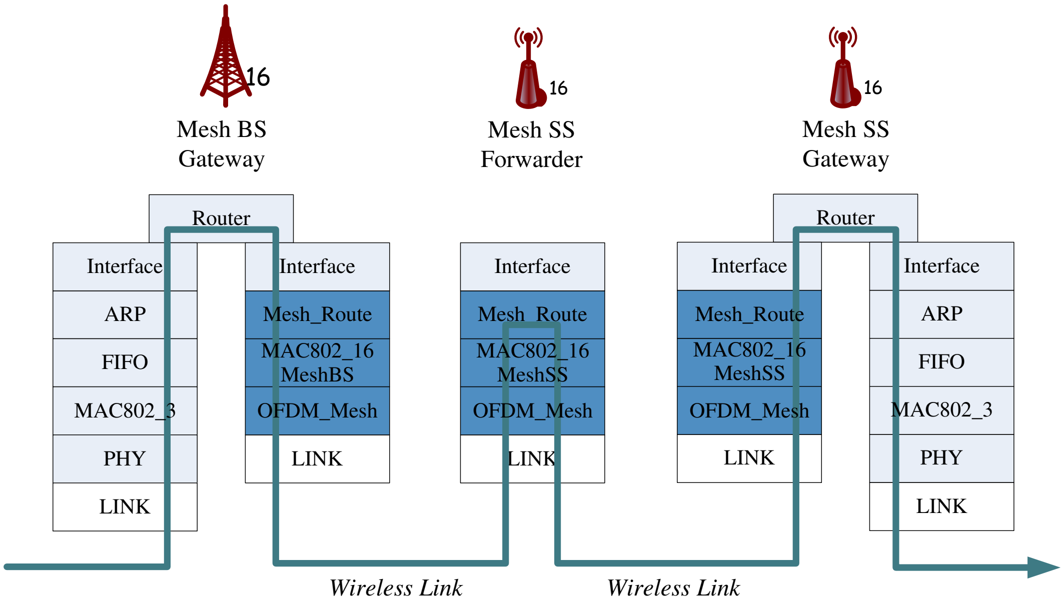


Figure 3.4: A forwarding example of packet processing flows

3.2 MAC Layer Design and Implementation

3.2.1 Mesh SS

Here we present the detailed design and implementation of our protocol module *MAC802_16_MeshSS*.

Network Entry Process

To support the network entry process in our simulation system, we design a network entry subsystem in *MAC802_16_MeshSS*. This subsystem includes procedures performed by the New Node, the Sponsoring Node, and the Registration Node. The state transition of the Sponsoring Node and the New Node in this process are shown in Fig. 3.5 and Fig. 3.6 respectively..

During the network entry process, some MAC messages are exchanged between nodes separated by multiple hops. For example, the REG-REQ and REG-RSP message are exchanged between the Registration Node and the New Node. (In our implementation, the Registration Node is co-located with the Mesh BS.) The Sponsoring Node is responsible for tunneling the REG-REQ received from the New Node

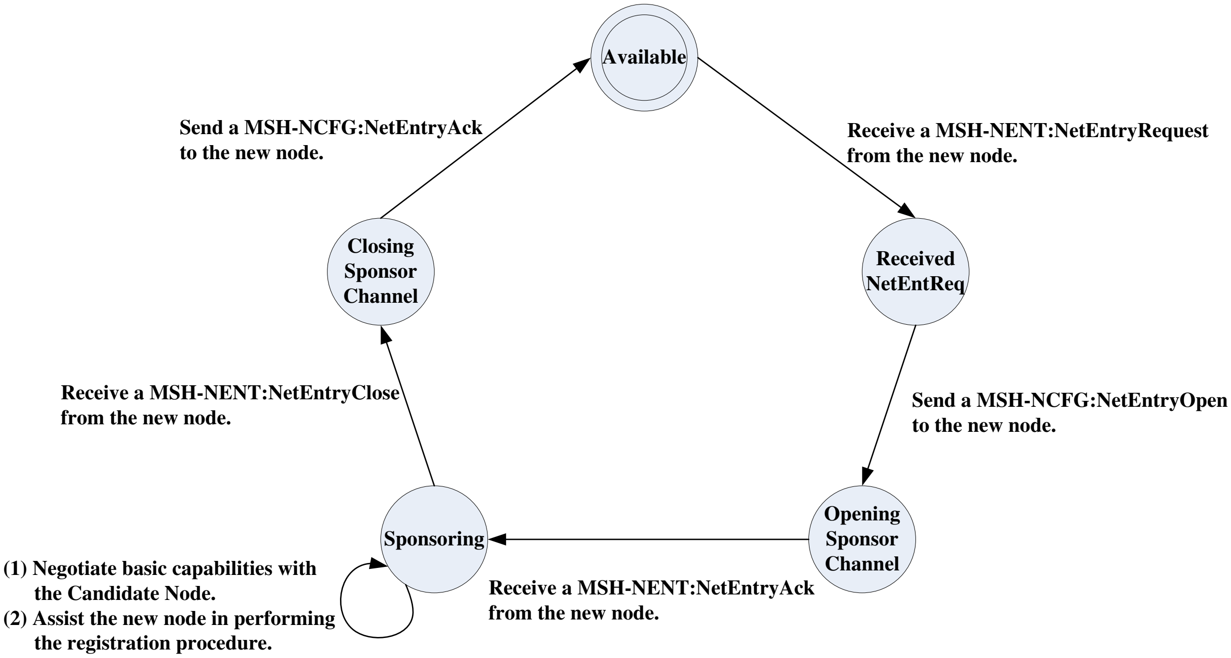

Available
Send a MSH-NCFG:NetEntryAck to the new node.
Receive a MSH-NENT:NetEntryRequest from the new node.
Closing Sponsor Channel
Received NetEntReq
Receive a MSH-NENT:NetEntryClose from the new node.
Send a MSH-NCFG:NetEntryOpen to the new node.
Sponsoring
Opening Sponsor Channel
(1) Negotiate basic capabilities with the Candidate Node.
(2) Assist the new node in performing the registration procedure.
Receive a MSH-NENT:NetEntryAck from the new node.

Figure 3.5: State transition of the Sponsoring Node

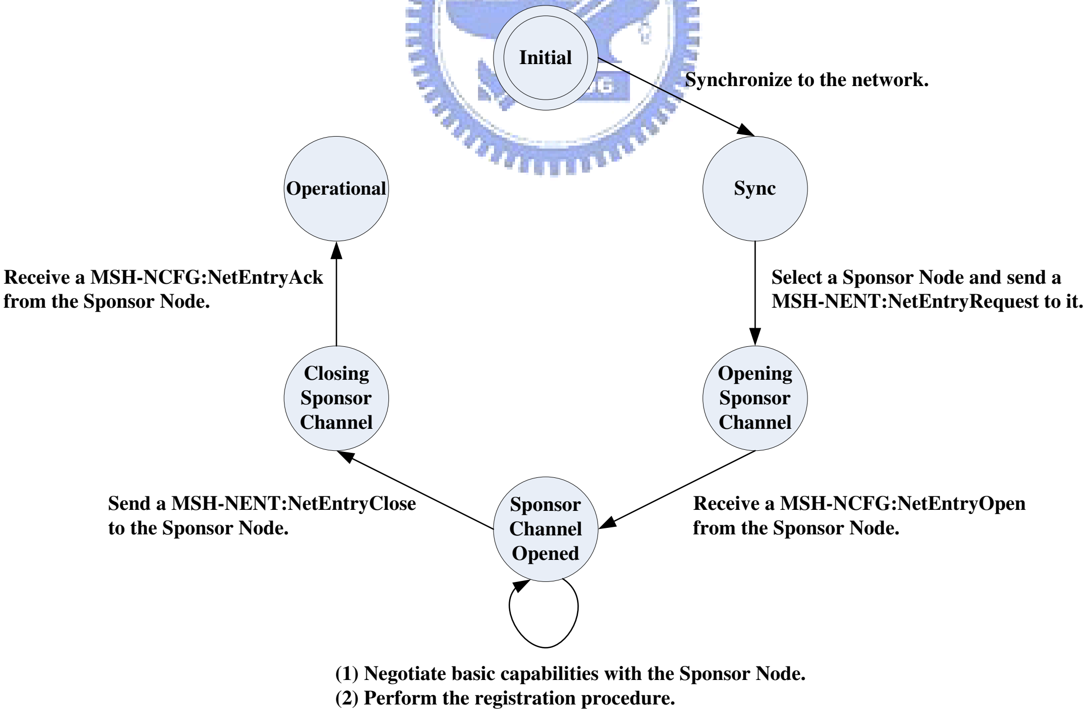

Initial
Synchronize to the network.
Operational
Sync
Receive a MSH-NCFG:NetEntryAck from the Sponsor Node.
Select a Sponsor Node and send a MSH-NENT:NetEntryRequest to it.
Closing Sponsor Channel
Opening Sponsor Channel
Send a MSH-NENT:NetEntryClose to the Sponsor Node.
Sponsor Channel Opened
Receive a MSH-NCFG:NetEntryOpen from the Sponsor Node.
(1) Negotiate basic capabilities with the Sponsor Node.
(2) Perform the registration procedure.

Figure 3.6: State transition of the New Node

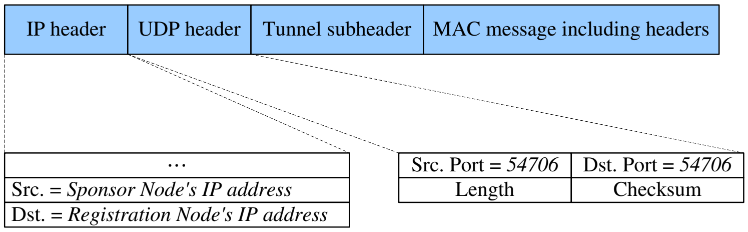

IP header
UDP header
Tunnel subheader
MAC message including headers
...
Src. = Sponsor Node's IP address
Dst. = Registration Node's IP address
Src. Port = 54706
Dst. Port = 54706
Length
Checksum

Figure 3.7: Format of tunneled messages

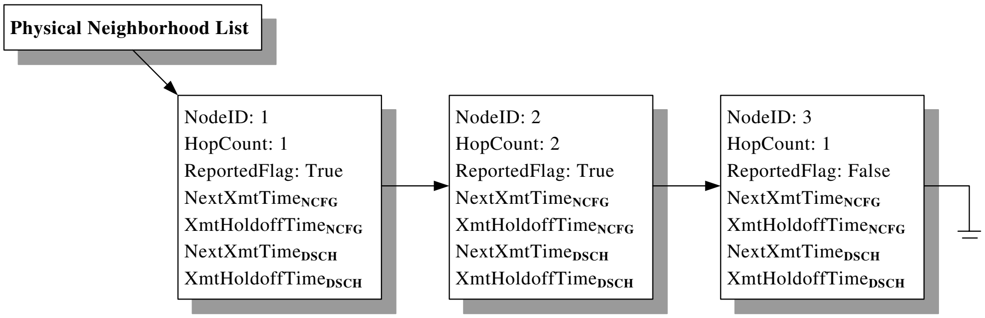


Figure 3.8: Physical neighborhood list

and forwarding the tunneled message to the Registration Node. Upon reception of the tunneled REG-RSP message from the Registration Node, the Sponsoring Node extracts the REG-RSP message and forwards it to the New Node. The tunneling mechanism is illustrated in Fig. 3.7. We use a specific port number (i.e. 54706) as an identifier so that the MAC layer modules can be aware of this action and respond accordingly.

Transmission Timing of Control Messages

In the Mesh network, basic functions including distributed scheduling and network synchronization are based on the physical neighbor list maintained by each node. Fig. 3.8 shows an example of the list. For the aforementioned functions to work correctly, each node shall regularly update the information contained in the neighborhood list. However, the IEEE 802.16 standard does not define how the information is updated.

Consider a simple network topology in Fig. 3.9. Two ambiguous situation may occur as follows:

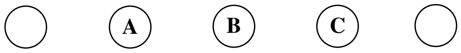


Figure 3.9: A simple chain topology

Case 1:

- At time t, Node A computes its scheduling information (**Next Xmt Mx** and **Xmt Holdoff Exponent**) based on the **Current Xmt Time** t. This information of Node A and its on-hop neighbors is included in its outgoing MSH-DSCH message. Node B, upon receiving this message, is able to determine the **Next Xmt Time** for Node A base on the **Current Xmt Time** t.

- At time $t + k$, Node B computes its scheduling information based on the **Current Xmt Time** $t + k$. This information of Node B and its one-hop neighbors is included in its outgoing MSH-DSCH message. Upon receiving the message, Node C computes a erroneous **Next Xmt Time** for Node A, since Node C has no idea when Node A transmits its last MSH-DSCH message.

Case 2:

- At time t, Node A computes its scheduling information and adjusts the information of its one-hop neighbors based on the **Current Xmt Time** t. All the information is included in its outgoing MSH-DSCH message. Node B, upon receiving this message, is able to determine the **Next Xmt Time** for Node A base on the **Current Xmt Time** t.

- At time $t+k$, Node B computes its scheduling information and adjusts the information of its one-hop neighbors based on the **Current Xmt Time** $t+k$. All the information is included in Node B's outgoing MSH-DSCH message. However, the new **Next Xmt Time** of Node A may not exactly match the original **Next Xmt Time** of Node A, if k is not a multiple of $2^{XmtHoldoffExponent_A}$ TxOpps. Therefore, upon receiving the message, Node C computes a erroneous **Next Xmt Time** for Node A accordingly.

In both cases, the erroneous information computed by Node C is likely to cause collisions in the succeeding control message exchanges, since Node C cannot determine the eligibility of Node A correctly. We can observe that to avoid the ambiguity, *a reference TxOpp number* is needed for both the sender and the receiver of the MSH-DSCH control messages. Therefore, we propose an amendment to the standard, *which requires no modification to any control message format or any existing mechanism*. In the following we describe our proposed amendment.

When a node is about to transmit its MSH-DSCH message, it first computes two reference TxOpp numbers by:

$$\begin{aligned}
Interval &= 2^{Xmt\,Holdoff\,Exponent} \\
Ref_1 &= \lfloor \frac{Current\,Xmt\,Time - 1}{Interval} \rfloor \cdot Interval + 1 \\
Ref_2 &= \lfloor \frac{Next\,Xmt\,Time - 1}{Interval} \rfloor \cdot Interval + 1
\end{aligned}$$

The Node's **Next Xmt Mx** can then be computed as follows:

$$Next\,Xmt\,Mx = \frac{Ref2 - Ref1}{Interval}$$

Upon receiving the MSH-DSCH message, its one-hop neighbors can compute its **Next Xmt Time** by

$$\begin{aligned}
Ref &= \lfloor \frac{Current\,Xmt\,Time - 1}{Interval} \rfloor \cdot Interval + 1 \\
Next\,Xmt\,Time\,Start &= Ref + Next\,Xmt\,Mx \cdot Interval \\
Next\,Xmt\,Time\,End &= Next\,Xmt\,Time\,Start + Interval - 1
\end{aligned}$$

The scheduling information of all its one-hop neighbors can also be adjusted using the above equations. Since all regular nodes in the Mesh network maintain the same TxOpp sequence number, the aforementioned ambiguity can be avoided. The timing is illustrated in Fig. 3.10. Our amendment also applies to MSH-NCFG messages, since the mechanisms to determine the transmission timing of both messages are the same.

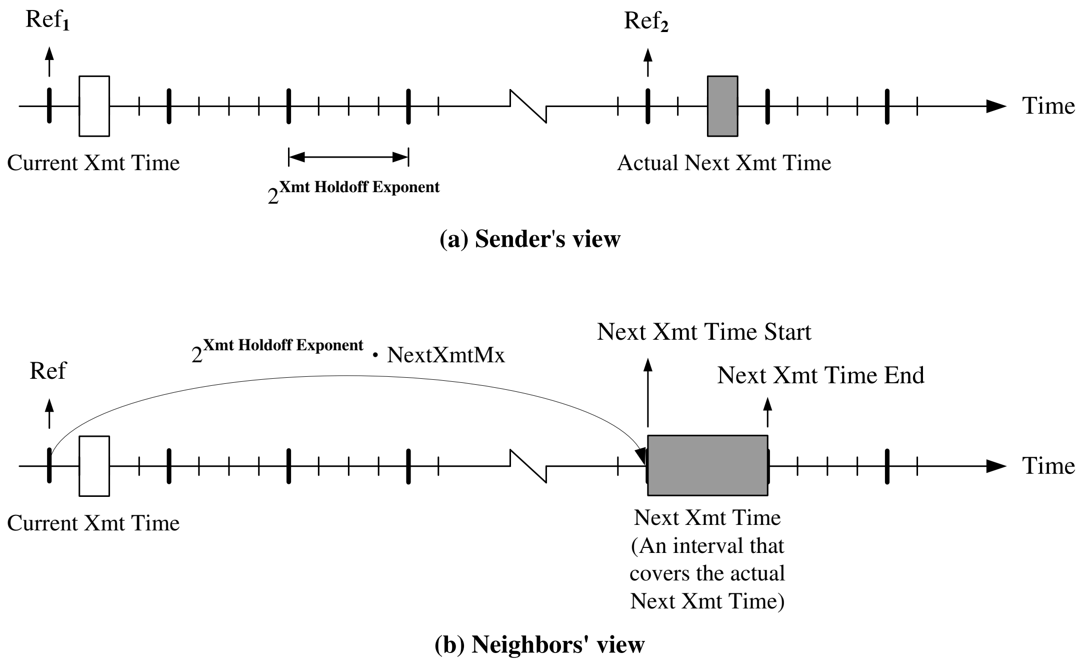


Figure 3.10: Our proposed amendment to the computation of control message transmission timing

Bandwidth Allocation and Request Mechanism

In the distributed scheduling mode, schedules are established between two nodes using the three-way handshake procedure as mentioned in Section 2.1.2. This procedure ensures the corresponding data transmissions are collision-free. However, it adversely induces a large delay to establish a schedule. Specifically, the requester suffers a delay of at least **Xmt Holdoff Time** TxOpps as shown in Fig. 3.11. This greatly reduces the link utilization and increases the latency of the packets sent by the upper layer protocol.

To overcome these problems, we propose an enhancement which allows the requester to have a chance to send a request *before its current schedule expires.* The used algorithm is listed in the next page. Chapter 5 evaluates the improvement made by our proposed enhancement.

The data subframe in which scheduled transmissions occur is divided into minislots. A minislot (or a range of minislots) may be available for requests or occupied by an established schedule. We design a state transition scheme for managing the

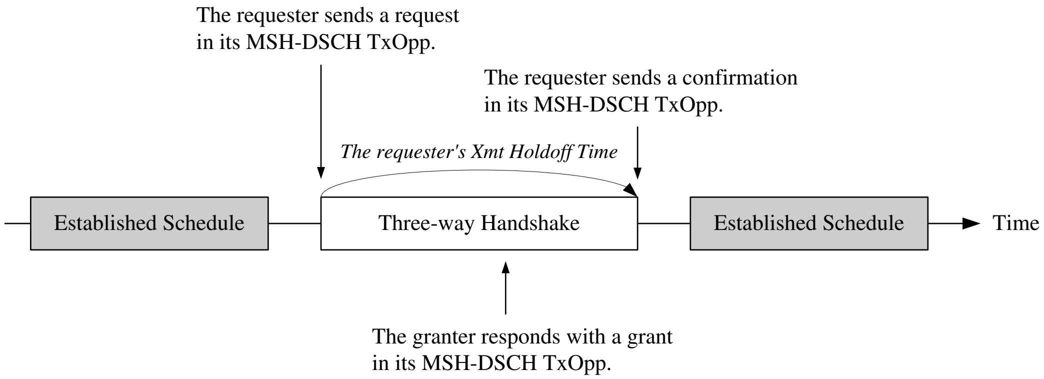


Figure 3.11: The delay induced by the three-way handshake procedure

Algorithm 1 Send Request

```
if there is no pending data then
    do nothing.
else if we don't have any schedules then
    append a request to the outgoing MSH-DSCH message.
else if we already have one schedule then
    $N \Leftarrow the\ frame\ number\ that\ the\ current\ schedule\ becomes\ valid$
    $L \Leftarrow the\ lifetime\ of\ the\ current\ schedule$
    if $N > current\ frame\ number$ then
        do nothing.
    else if $L > Xmt\ Holdoff\ Time$ then
        do nothing.
    else
        append a request to the outgoing MSH-DSCH message.
    end if
end if
```

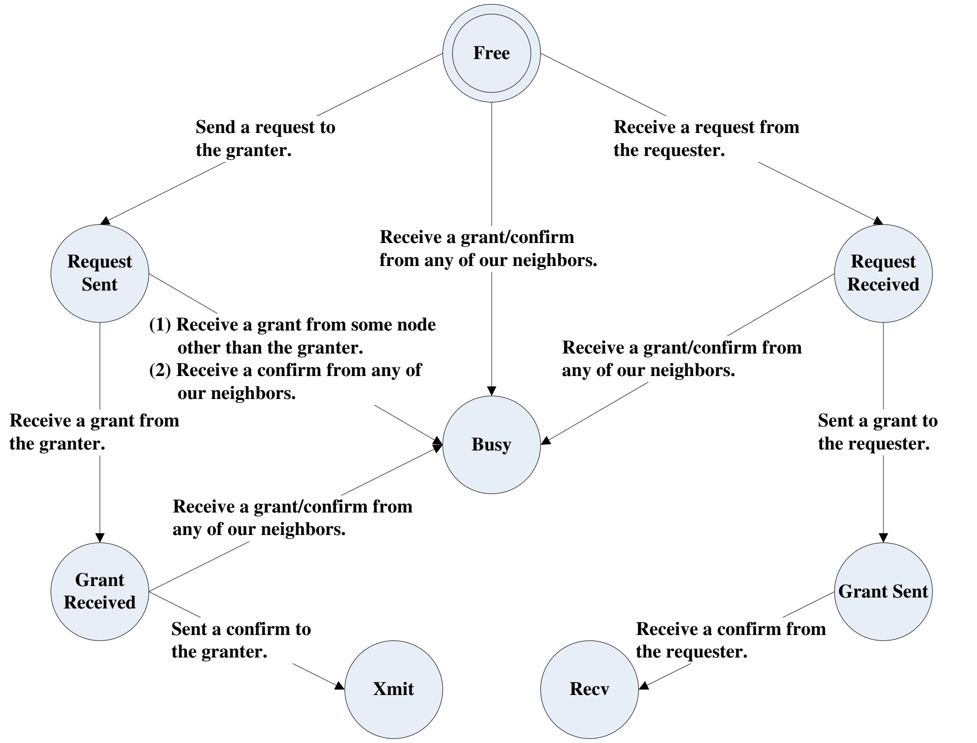

Free
Send a request to the granter.
Receive a request from the requester.
Request Sent
Receive a grant/confirm from any of our neighbors.
Request Received
(1) Receive a grant from some node other than the granter.
(2) Receive a confirm from any of our neighbors.
Receive a grant/confirm from any of our neighbors.
Receive a grant from the granter.
Busy
Sent a grant to the requester.
Receive a grant/confirm from any of our neighbors.
Grant Received
Grant Sent
Sent a confirm to the granter.
Receive a confirm from the requester.
Xmit
Recv

Figure 3.12: Minislot state transition

minislots. Fig. 3.12 illustrates the state transition of a minislot.

3.2.2 Mesh BS

The Mesh BS provides backhaul connectivity for the entire mesh network. The protocol stacks of the Mesh BS are shown in Fig. 3.2. It can be seen that the wired protocol stack represents a direct link to the Internet.

As mentioned earlier, in our implementation the Mesh BS also performs the registration function, i.e. the Node ID assignment. For this purpose, we generate a configuration file containing mappings of MAC addresses and Node IDs. When the Mesh BS receives a REG-REQ message from a New Node, it maps the MAC address carried in the message to a unique Node ID. The Node ID is included in the REG-RSP message and returned to the New Node.

Despite the two functions described above, using distributed scheduling, the Mesh BS is merely a peer node in the network. Therefore, it shall also assist new nodes in entering the mesh network, broadcast control messages periodically, and establish distributed schedule with its neighbors.

3.3 PHY Layer Design and Implementation

3.3.1 Channel Coding

We implement a rate $\frac{1}{2}$ systematic convolutional code that realizes various code rates with puncturing. A hard-decision Viterbi decoder is used to decode the coded bit stream. When the punctured pattern is used, dummy bits that do not affect the metric are inserted in the corresponding positions.

The randomizer and the interleaver are both implemented as defined in the standard. The Reed-Solomon code implementation is obtained from [7].

3.3.2 Channel Model

For simulating the wireless channel, we adopt an empirically based path loss model presented in [4] and [5]. The path loss (in dB) is

$$PL \;=\; A + 10 \cdot \gamma \cdot \log \frac{d}{d_0} + s + \Delta PL_f + \Delta PL_h$$

where A is a fixed quantity given by the free-space path loss formula at distance $d_0 = 100m$, s represents the shadow fading variation, and the path loss exponent γ is a Gaussian random variable given by

$$\gamma \;=\; (a - b \cdot h_b + \frac{c}{h_b}),\ 10m \geq h_b \geq 80m$$

where h_b is the base station antenna height in meters and a, b, c are constants dependent on the terrain category given in Table 3.1. Category A, the maximum path loss category, is a hilly terrain with moderate-to-heavy tree densities. Category B is either a mostly flat terrain with moderate-to-heavy tree densities, or a hily terrain with light tree densities. Category C, the minimum path loss category, is

Table 3.1: Numerical values of model parameters.

Model parameter	Category A	Category B	Category C
a	4.6	4.0	3.6
b	0.0075	0.0065	0.005
c	12.6	17.1	20.0

a mostly flat terrain with light tree densities. For different frequencies and receive antenna heights, the following correction terms are used:

$$\begin{aligned} \Delta PL_f &= 6 \cdot \log \frac{f}{2000} \\ \Delta PL_h &= -10.8 \cdot \log \frac{h}{2}; \quad for\ Categories\ A\ and\ B \\ \Delta PL_h &= -20 \cdot \log \frac{h}{2}; \quad for\ Category\ C \end{aligned}$$

where f is the freqyency in MHz, and h is the receive antenna height between 2m and 10m.

We adopt Rayleigh fading channels, because WiMAX Mesh networks are likely to be deployed in non-LOS environments. The bit-error-rates (BERs) of various modulation modes under Rayleigh fading channels are given by:

$$\begin{aligned} P_{BPSK/QPSK} &= \frac{1}{2}(1 - \sqrt{\frac{\gamma}{1+\gamma}}) \\ P_{QAM} &= 2(\frac{\sqrt{M}-1}{\sqrt{M}})\frac{1}{\log_2 M}\sum_{i=1}^{\sqrt{M}/2}(1 - \sqrt{\frac{1.5(2i-1)^2\gamma\log_2 M}{M-1+1.5(2i-1)^2\gamma\log_2 M}}) \end{aligned}$$

where γ is the energy per bit per noise power spectral density (i.e. $\frac{E_b}{N_0}$), and M is 16 for 16-QAM and 64 for 64-QAM.

3.4 A Simple Route Module

In our current implementation, we adopt a *fixed* layer-3 routing scheme, which can be later replaced by other layer-3 routing protocols such as AODV, OSPF and RIP.

Table 3.2: An example of the fixed routing table.

Owner	**IP address/Netmask**	**Next-hop Node ID**
$node_(1)	1.0.1.2/32	2
$node_(1)	1.0.1.3/32	2
$node_(2)	1.0.1.1/32	1
$node_(2)	1.0.1.3/32	3
$node_(3)	1.0.1.1/32	2
$node_(3)	1.0.1.2/32	2

The fixed routing table is generated and stored in a file before the simulation starts. An example of the routing table is shown in Table 3.2.

Chapter 4

Functionality Validation

4.1 Network Entry Process

To validate our implementation of the network entry process, we demonstrate an simulation example step by step. Fig. 4.1 shows the network topology with a Mesh BS located in Node 1. The steps of the network entry process to the stage when all nodes can start scheduled transmissions are as follows:

1. Node 1 periodically broadcasts MSH-NCFG messages.

2. Node 2 and Node 3 synchronize with Node 1, build their physical neighbor lists in the meantime, and select Node 1 as their Sponsoring Node.

3. Node 2 and Node 3 transmit MSH-NENT messages with network entry requests to Node 1 at the same time, causing a collision at Node 1. Neither of

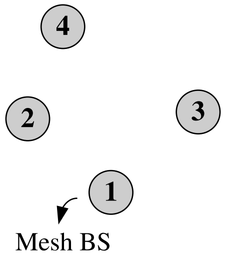


Figure 4.1: The simulation scenarios used in Section 4.1

the messages is successfully received by Node 1.

4. Node 2's and Node 3's timers set for receiving responses from the Sponsoring Node expire. They both defer their further attempts according to a random backoff mechanism.

5. The deferring time of Node 3 is up.

6. Node 3 transmits a MSH-NENT message with a network entry request to Node 1. Upon reception of this message, Node 1 responds to Node 3 by transmitting a MSH-NCFG message with a network entry grant to Node 3. The Sponsor Channel is established after Node 3 transmits a MSH-NENT message with a network entry acknowledgment to Node 1.

7. After basic capabilities are negotiated and authorization is performed, Node 3 registers with the Mesh BS (i.e. Node 1) by transmitting a REG-REQ message to it. The Mesh BS then replies a REG-RSP message with a unique Node ID assigned to Node 3.

8. The Sponsor Channel is closed and Node 3 becomes a regular node in the network.

9. Node 2's deferring time is up and repeats step 6 to step 8.

10. Node 4 synchronizes with Node 2, builds its physical neighborhood list in the meantime, and selects Node 2 as its Sponsoring Node. Step 6 to step 8 are repeated except that (1) the REG-REQ message transmitted by Node 4 is tunneled by Node 2 and forwarded to the Mesh BS, and (2) the REG-RSP message transmitted by the Mesh BS is extracted by Node 2 and delivered to Node 4.

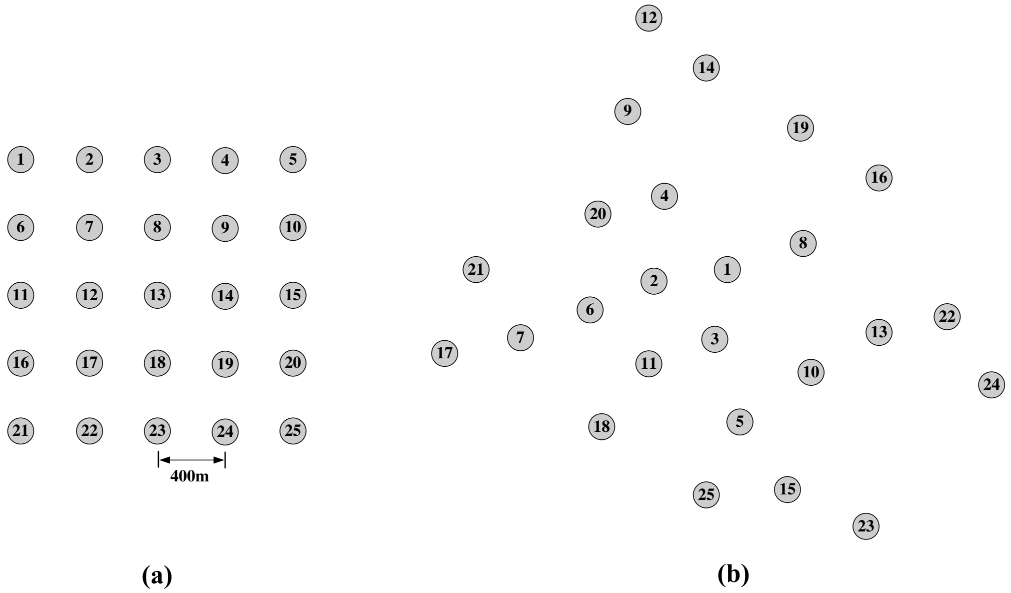


Figure 4.2: The simulation scenarios used in Section 4.2: (a) a 5x5 grid topology, and (b) a general topology consisting of 25 nodes

4.2 Distributed Election-based Scheduling

As mentioned in Section 2.1.2, the transmission timing of MSH-DSCH and MSH-NCFG messages is determined by the distributed EBTT mechanism. In the following we examine our implementation of this mechanism by simulations. The simulation scenarios are shown in Fig. 4.2, each consisting of 25 nodes. Table 4.1 lists important network parameters used in both scenarios. Because it takes some time for the network to become stable (i.e. all nodes have completed the network entry process), we start collecting simulation results at 80 seconds.

Fairness

In the long term, all nodes should acquire an *almost* equal number of TxOpps if they have the same value of **Xmt Holdoff Exponent**. This fairness is ensured by the way the pseudorandom mixing number is computed.

Fig. 4.3 shows the TxOpps acquired by each node in the simulation case of the 5x5 grid topology. We see that nodes located in the center of the grid, for example

Table 4.1: Simulation parameters used in Section 4.2

Parameter name	**Value**	**Description**
Xmt Holdoff Exponent	1	Xmt Holdoff Time = $2^{1+4} = 32$ TxOpps
MSH-CTRL-LEN	8	TxOpps per frame
MSH-DSCH-NUM	8	MSH-DSCH TxOpps per frame
Scheduling-Frames	1	Four frames have a schedule control subframe between two frames with network control subframes
Frame-Duration	10 ms	
Simulation Time	400 sec	Simulation results are collected after 80 seconds.

node Node 12, 13 and 14, have slightly less TxOpps than their neighboring nodes. This phenomenon can be explained because nodes with more neighbors within their extended neighborhood have more potential competitors when the Mesh Election for some TxOpp is held. A similar result is obtained for the general topology case, which is shown in Fig. 4.4.

Latency

The latency of a node is the number of TxOpps between its two consecutive MSH-DSCH/MSH-NCFG transmissions. Intuitively, the average latency is inversely proportional to the number of TxOpps acquired. Moreover, a node's latency should approximate to its **Xmt Holdoff Time** if the number of neighbors within its extended neighborhood does not exceed its **Xmt Holdoff Time**.

The average latency experienced by each node is illustrated in Fig. 4.5. All nodes have an average latency slighly more than the **Xmt Holdoff Time**. A similar result is obtained for the general topology case, which is shown in Fig. 4.6.

Exclusion

Within the extended neighborhood, only one node is permitted to transmit at a time. This ensures that the exchange of MSH-DSCH and MSH-NCFG messages is

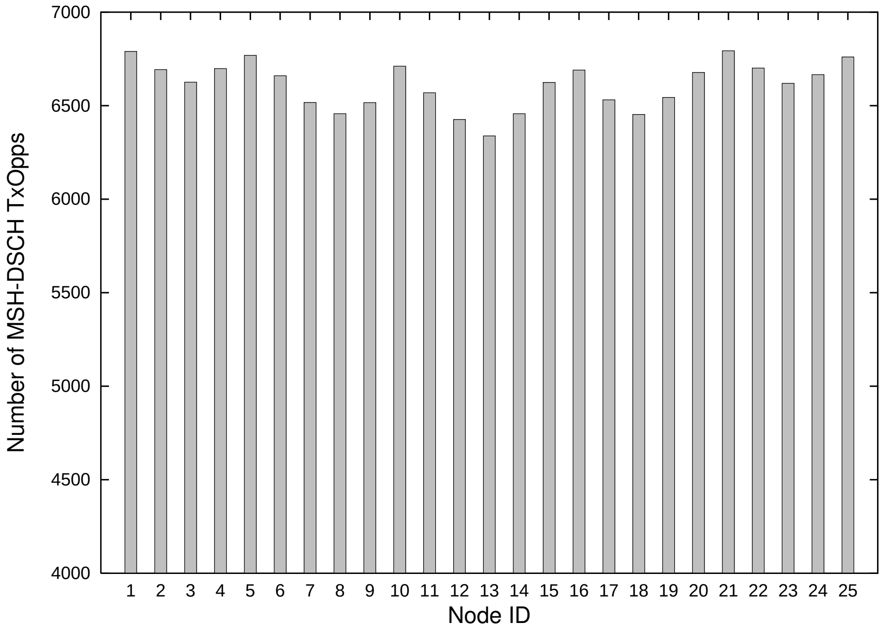


Figure 4.3: Number of MSH-DSCH TxOpps vs. Node ID for the network topology in Fig. 4.2(a)

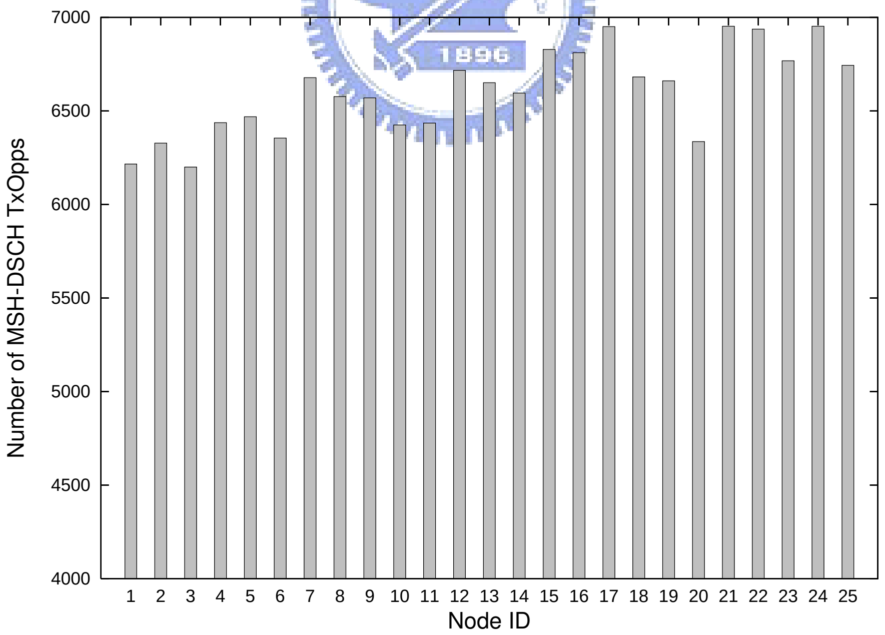


Figure 4.4: Number of MSH-DSCH TxOpps vs. Node ID for the network topology in Fig. 4.2(b)

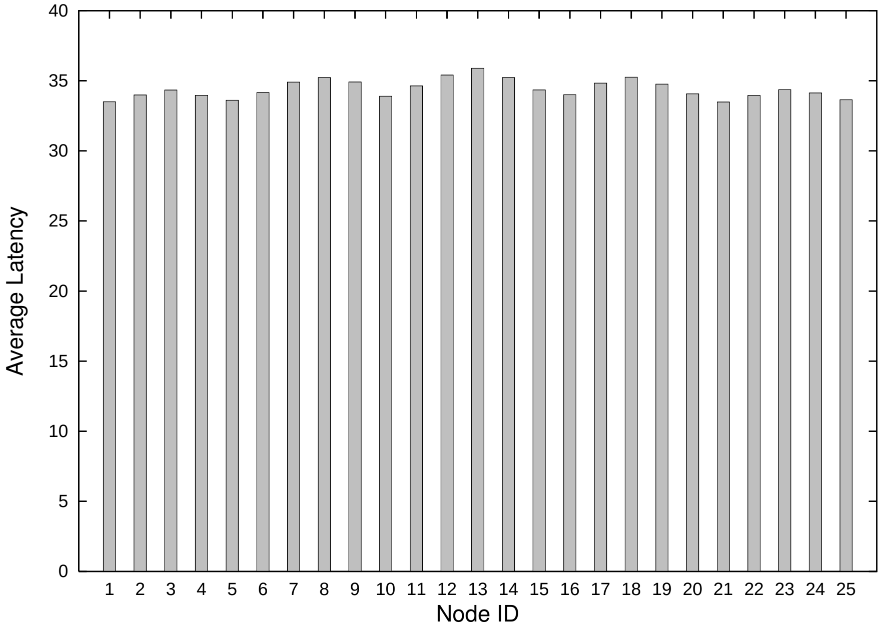


Figure 4.5: Average latency vs. Node ID for the network topology in Fig. 4.2(a)

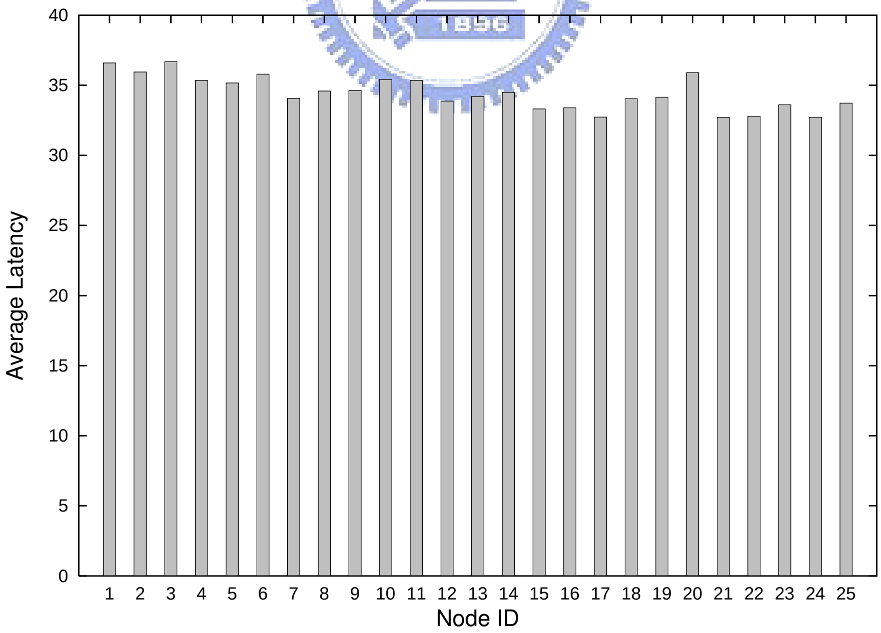


Figure 4.6: Average latency vs. Node ID for the network topology in Fig. 4.2(b)

Table 4.2: Simulation parameters used in Section 4.3

Parameter name	**Value**	**Description**
Tb	11.1 us	OFDM useful symbol time
Tg	Tb/4	OFDM guard time
MSH-CTRL-LEN	8	TxOpps per frame
Frame Duration	10 ms	

collision-free. The simulation results of the above two topologies show that there is no collision after the network becomes stable.

4.3 PHY Modes and Corresponding Throughput

In this section, we compute the theoretical capacity at the PHY layer and the MAC layer using parameters listed in Table 4.2. The derived throughput is then compared with the application throughput obtained from our simulation results.

Let $T_{Ctrl.}$ and T_{Data} denote the throughput of the control subframe and the data subframe, respectively; S_{Frame} denotes the number of OFDM symbols per frame; $S_{Ctrl.}$ and S_{Data} denote the number of OFDM symbols in the control subframe and the data subframe, respectively. The PHY throughput, T_{PHY}, over a link can be expressed as:

$$T_{PHY} = T_{Ctrl.} + T_{Data}$$

There are MSH-CTRL-LEN TxOpps per control subframe, with each TxOpp consisting of 7 OFDM symbols. So we have

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

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

$$= \quad 664$$

All transmissions in the control subframe are sent using the QPSK-1/2 mode. The size of a coded block in this mode is 48 bytes. Thus, $T_{Ctrl.}$ is a fixed value and can be computed as :

$$\begin{aligned} T_{Ctrl.} &= \frac{S_{Ctrl.} \cdot 48}{Frame\,Duration} \\ &= 268.8\,Kbyte/sec \\ &= 2.15\,Mbit/sec \end{aligned}$$

T_{Data} depends on the type of the PHY mode used. For example, when operating in the QPSK-3/4 mode, T_{Data} can be computed as:

$$\begin{aligned} T_{Data} &= \frac{S_{Data} \cdot 48}{Frame\,Duration} \\ &= 3.19\,Mbyte/sec \\ &= 25.5\,Mbit/sec \end{aligned}$$

Using the same example, the PHY throughput is $2.15 + 25.5 = 27.65 Mbit/sec$. The MAC throughput, T_{MAC}, can be derived by considering the coding rate:

$$\begin{aligned} T_{MAC} &= T_{Ctrl.} \cdot \frac{1}{2} + T_{Data} \cdot \frac{3}{4} \\ &= 20.2\,Mbit/sec \end{aligned}$$

The PHY throughput and the MAC throughput using mandatory PHY modes are shown in the second and the third columns of Table 4.3. To obtain the application throughput, we setup a simple simulation case containing only two mesh nodes. One of the nodes is running a greedy UDP sender program, transmitting data packets as soon as possible, while the other is running a UDP receiver program. To fully utilize the available bandwidth, a permanent distributed schedule which spans the whole data subframe is established between the sender and the receiver. The simulation results are shown in the fourth column of Table 4.3. Protocol overheads such

Table 4.3: PHY, MAC and application throughput using mandatory PHY modes

Mandatory	**PHY**	**MAC**	**Application**
mode	**throughput**	**throughput**	**throughput**
	(Mbit/sec)	**(Mbit/sec)**	**(Mbit/sec)**
BPSK-1/2	14.90	7.45	6.14
QPSK-1/2	27.65	13.82	12.35
QPSK-3/4	27.65	20.2	18.45
16QAM-1/2	53.15	26.57	24.61
16QAM-3/4	53.15	39.32	36.92
64QAM-2/3	78.64	52.07	49.23
64QAM-3/4	78.64	58.44	55.39

as UDP/IP headers, the MAC header/subheaders/trailer (CRC), and MAC management messages exchanged in the control subframe contribute to the difference between the MAC throughput and the application throughput.

Chapter 5

Performance Evaluation

In this chapter, we evaluated the performance of the WiMAX Mesh network. Throughout this chapter, the common simulation parameters listed in Table 5.1 are used. The channel model in the physical layer is disabled by default. The MAC throughput in the data subframe, $T_{MAC,\,Data}$, over a link is

$$\begin{aligned} T_{MAC,\,Data} &= \frac{Request\,Size \cdot Minislot\,Size \cdot Uncoded\,Block\,Size}{Frame\,Duration} \\ &= \frac{10 \cdot 3 \cdot 108}{10\,ms} \\ &= 324.0\,Kbyte/sec \end{aligned}$$

5.1 Multihop Traffic

The IEEE 802.16 Mesh mode is primarily defined to support multihop wireless communication. However, when distributed scheduling is used, the three-way handshake procedure to establish schedules between two nodes incurs an unavoidable large delay. Therefore, as mentioned in Section 3.2.1, we propose a scheme for Mesh nodes to aggressively establish distributed schedules with their neighbor before their current schedules are expired. In the following, we perform two suites of performance tests to observe the performance of UDP/TCP multihop connections over the WiMAX Mesh network. We also evaluate the improvement made by our proposed scheme.

Table 5.1: Common simulation parameters

Parameter name	**Value**	**Description**
MSH-CTRL-LEN	8	TxOpps per frame
Xmt Holdoff Exponent	1	Xmt Holdoff Time = 2^{1+4} = 32 TxOpps
Schedule Persistence	128	Number of frames over which the schedule is valid
Request Size	10	Maximal number of minislots per established schedule
Minislot Size	3	Number of OFDM symbols per minislot
PHY mode	64QAM-2/3	108 bytes per uncoded block
Frame Duration	10 ms	

The network configuration used is shown in Fig. 5.1. The node on the left hand side is the source host while the host on the right hand side is the destination host. Intermediate nodes act as traffic forwarders. In the first suite, an greedy UDP connection is established between the source and the destination. In the second suite, a greedy TCP connection is established instead. The simulation time of each suite is 300 seconds.

Fig. 5.2 shows the simulation results of the first test suite. We see that for the 1-hop case without using our proposed scheme, there is a gap between $T_{MAC,\,Data}$ and the UDP application throughput. The reasons are twofold. First, various unavoidable protocol overheads contribute to this gap. Second, the sender issues a request *only after the current schedule is expired.* Therefore, there is a delay before the new schedule can be established. During the schedule setup time, no application

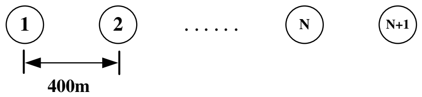


Figure 5.1: The network topology used to evaluate the performances of UDP/TCP multihop connections

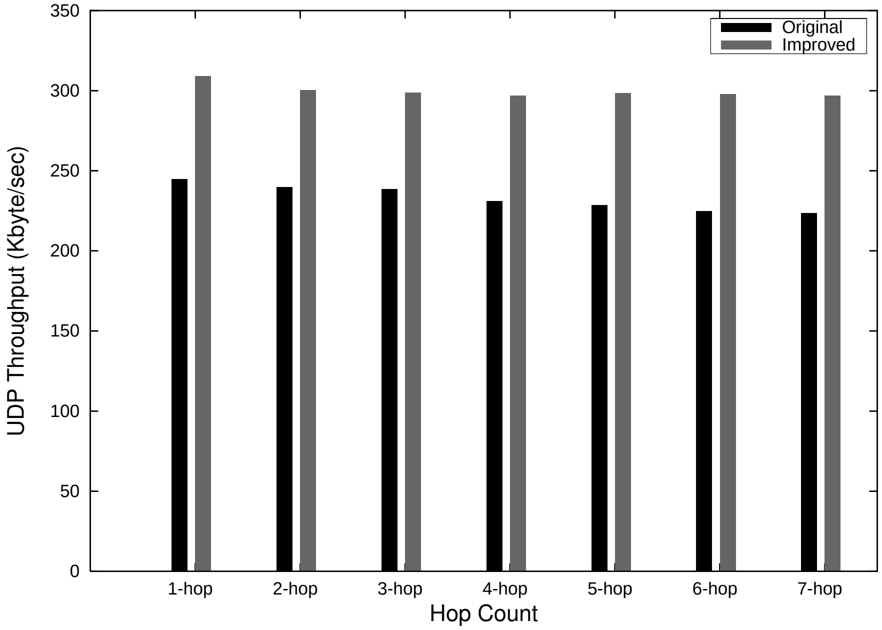


Figure 5.2: Performances of multihop UDP connections

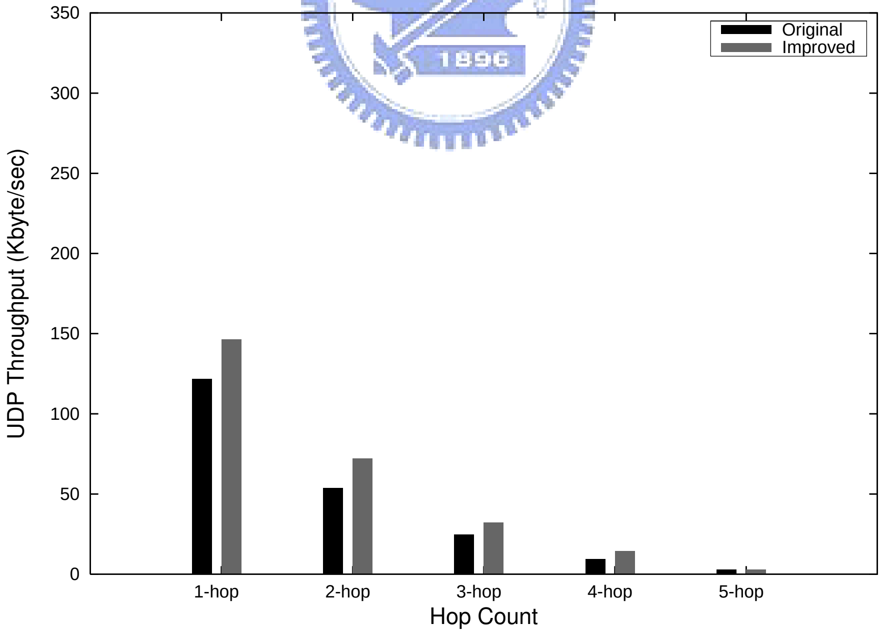


Figure 5.3: Performances of multihop UDP connections under a more realistic wireless channel

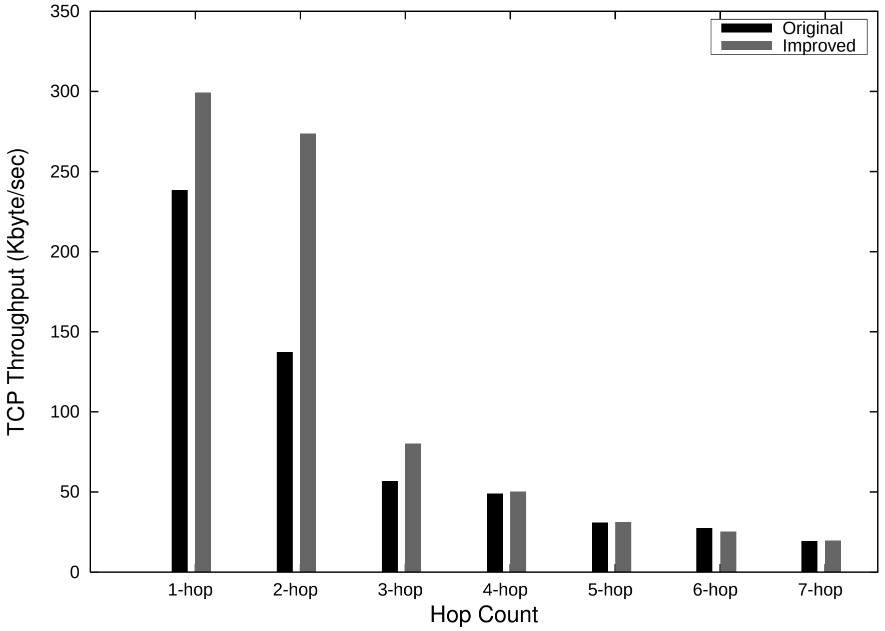


Figure 5.4: Performances of multihop TCP connections

data can be transmitted. It can be seen that using our proposed scheme there is a constant improvement of about 25%, because the delay has been mostly eliminated. Fig. 5.3 shows the simulation results of the same test suite with the channel model enabled. Unlike Fig. 5.2, performance of multihop UDP connections downgrade as the number of hops increases.

Fig. 5.4 shows the simulation results of the second test suite. For 1-hop, 2-hop and 3-hop TCP connections, our proposed scheme improve the TCP application throughput by 25%, 99% and 41%, respectively. Note that the performance of TCP connections are constrained by the round-trip-time (RTT) between the source and the destination. Our proposed scheme can only eliminate the schedule setup delay from the source to the destination. In the reverse direction, however, it cannot be triggered since TCP ACK packets are sent in a much slower rate.

5.2 TCP Fairness

TCP is one of the most important protocols of the Internet protocol suite. Many popular applications including WWW and E-mail are built on top of TCP. It is a connection-oriented protocol and guarantees reliable and in-order delivery over the connections established between end systems. To achieve high performance without adding the network's level of congestion, several techniques including slow-start, flow control and congestion control are used. In this section, we set up a simulation case to demonstrate that TCP congestion control exhibits fairness when sharing the same bottleneck wireless links in the WiMAX Mesh network.

The network configuration used is depicted in Fig. 5.5. In this configuration, there are six nodes, one Mesh BS, two Mesh SSs, two networked hosts connected to the Mesh BS gateway, and one networked host connected to the Mesh SS Gateway. Two greedy TCP connections are established between the two TCP senders and the TCP receiver. The simulation time is 100 seconds.

Fig. 5.6 shows the simulation result of this case. It can be seen that TCP still exhibits long-term fairness across the WiMAX Mesh network.

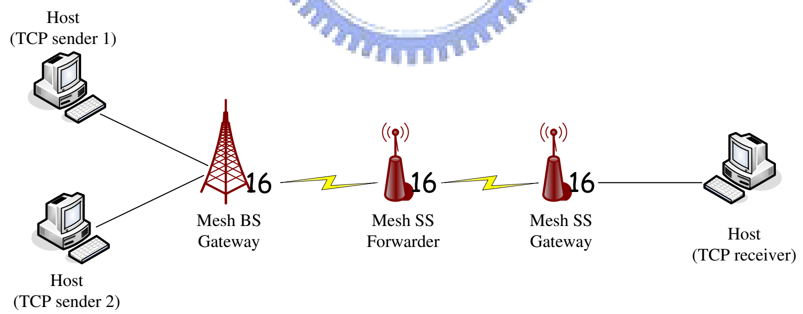


Figure 5.5: The network topology used to demonstrate TCP fairness over WiMAX Mesh network links

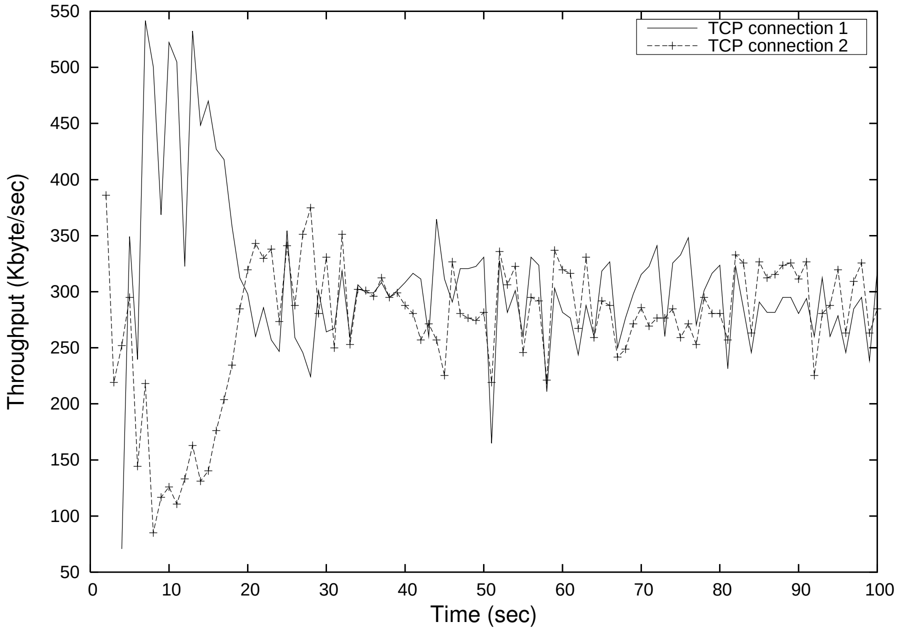


Figure 5.6: TCP fairness over WiMAX Mesh network links

5.3 Downstream Traffic

When devices of the IEEE 802.16 Mesh Mode come onto the market, they will be mainly deployed in the metropolitan area, providing people with a new way to access the Internet. Typical Internet users create much more downstream traffic, for example browsing web pages, than upstream traffic. Therefore, in this section, we perform two suits of tests to assess the performance of downstream traffic in the WiMAX Mesh network.

The network scenarios shown in Fig. 5.7 are used for both suites. In the first suite, there are 24 UDP connections established between the Mesh BS(s) and each of the Mesh SSs to initiate the Internet downstream traffic pattern. In the second suite, TCP connections are established instead. The simulation time is 400 seconds for each suite and the traffic generators start sending data at 100 seconds. Each simulation is run for 10 times.

Simulation results of Fig. 5.7(a) in the first suite are shown in Fig. 5.8. The total system performance is 1587.72 Kbyte/sec. Using distributed scheduling, all Mesh nodes including the Mesh BS contend network resources in a peer-to-peer fashion. In

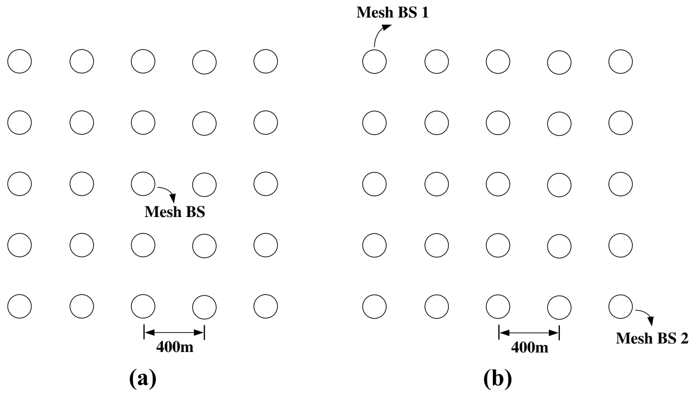


Figure 5.7: Network topologies for evaluating the downstream performance

other words, the Mesh BS has no higher priority over other nodes. As downstream traffic primarily comes from the Mesh BS, the wireless links between the Mesh BS and its surrounding Mesh SSs become a bottleneck of the entire system. Fig. 5.7(b) is an example of a multi-gateway configuration. Two Mesh BSs are deployed in two corners of the grid topology. The total system performance is 1908.16 kbyte/sec, as shown in Fig. 5.9. This configuration makes an improvement of 20% compared with Fig. 5.8.

Simulation results of the second suite are shown in Fig. 5.10 and Fig. 5.11. The total system performances are 1146.21 Kbyte/sec and 1673.47 kbyte/sec respectively. As discussed in Section 5.1, TCP throughput downgrades as the number of hops between the source node and the destination node increases.

5.4 Client-to-Client Traffic

Recently, peer-to-peer (P2P) computing has attracted a lot of interest from both the research area and the real world. More and more systems and applications are built using P2P technology, such as instant messaging, file sharing and VoIP services. Therefore, in this section, we perform two suits of tests to assess the performance of client-to-client traffic in WiMAX Mesh networks.

The network configuration used is a grid topology consisting of 25 nodes. In the first suite, there are 25 UDP connections established between two randomly selected nodes to initiate the P2P traffic pattern. In the second suite, TCP connections are established instead. The simulation time is 400 seconds for each suite and the traffic generators start sending data at 100 seconds. Each simulation is run for 10 times.

Simulation results are shown in Fig. 5.12 and Fig. 5.13. The total system performances are 2802.32 Kbyte/sec and 1601.82 Kbyte/sec respectively.

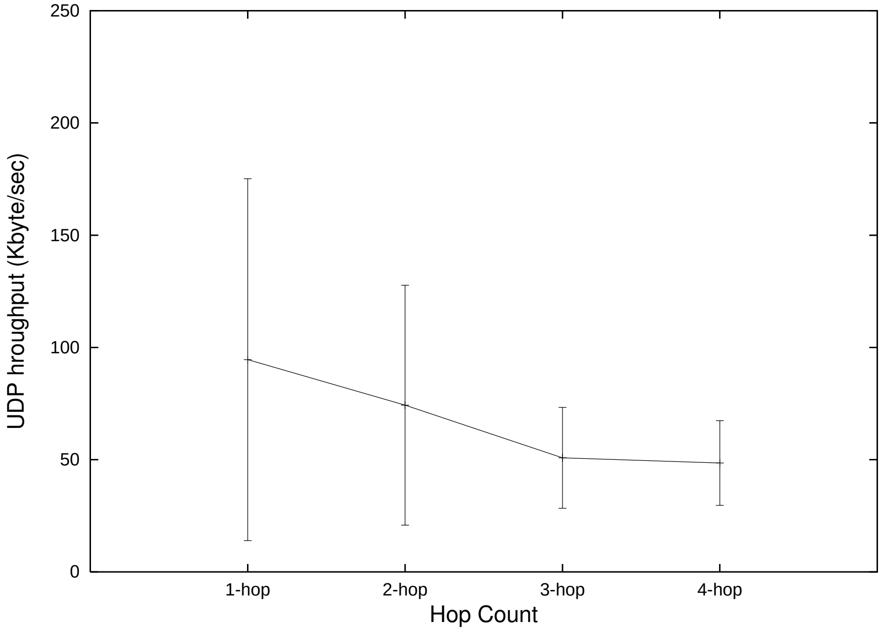


Figure 5.8: Downstream UDP performance for the network topology in Fig. 5.7(a)

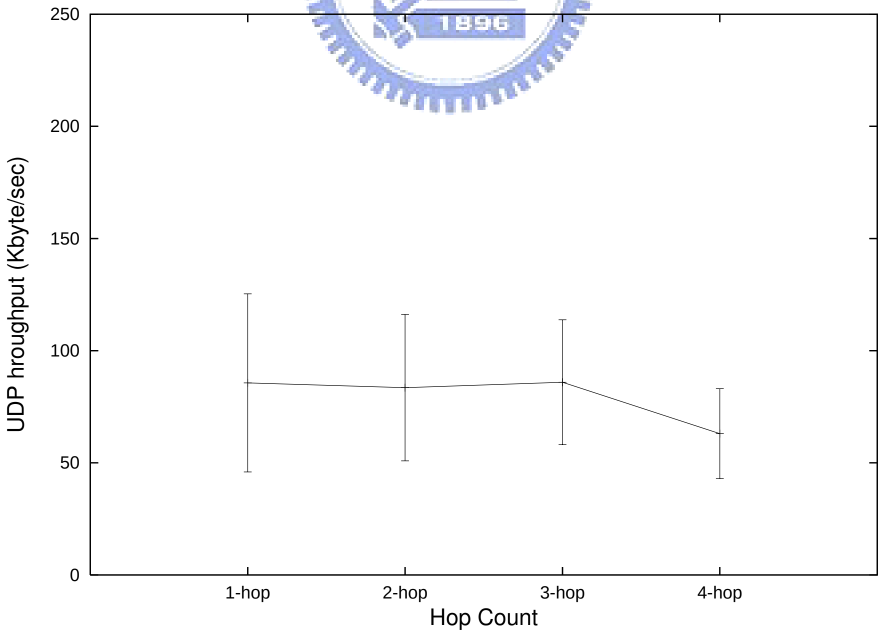


Figure 5.9: Downstream UDP performance for the network topology in Fig. 5.7(b)

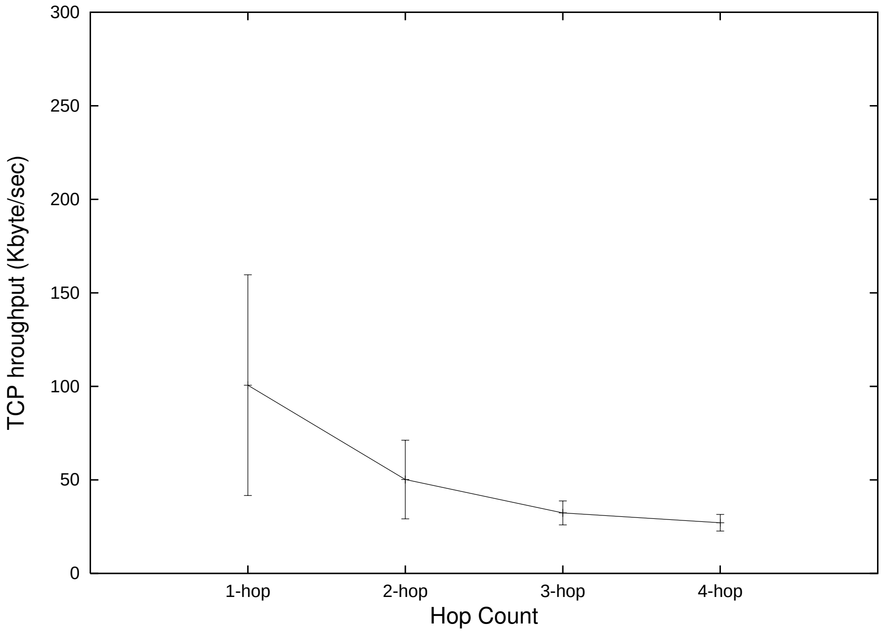


Figure 5.10: Downstream TCP performance for the network topology in Fig. 5.7(a)

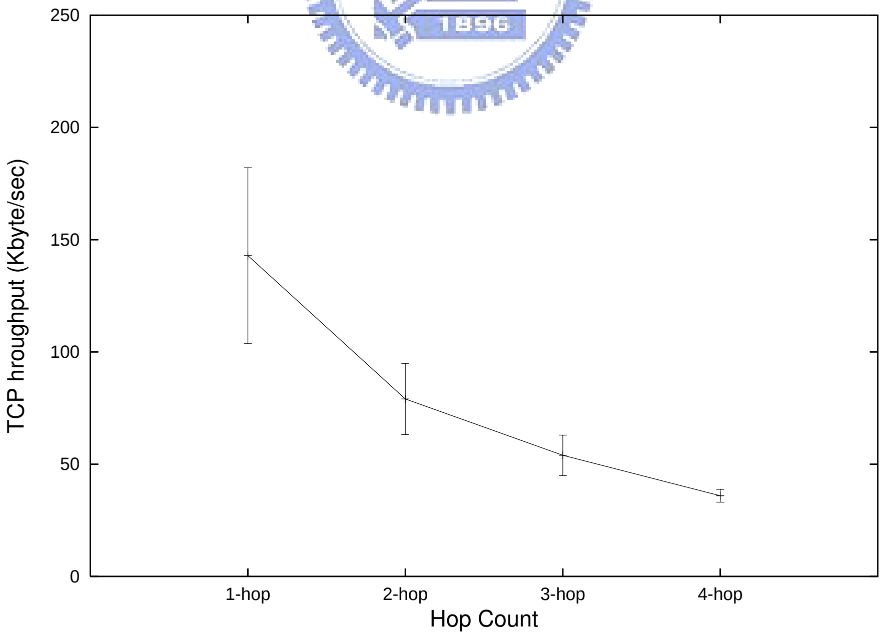


Figure 5.11: Downstream TCP performance for the network topology in Fig. 5.7(b)

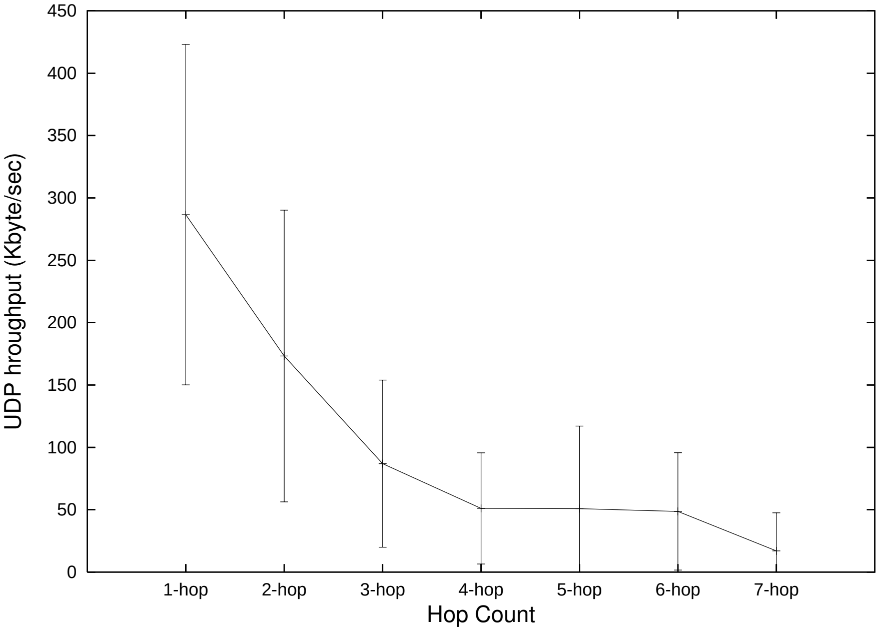


Figure 5.12: Client-to-client UDP performance

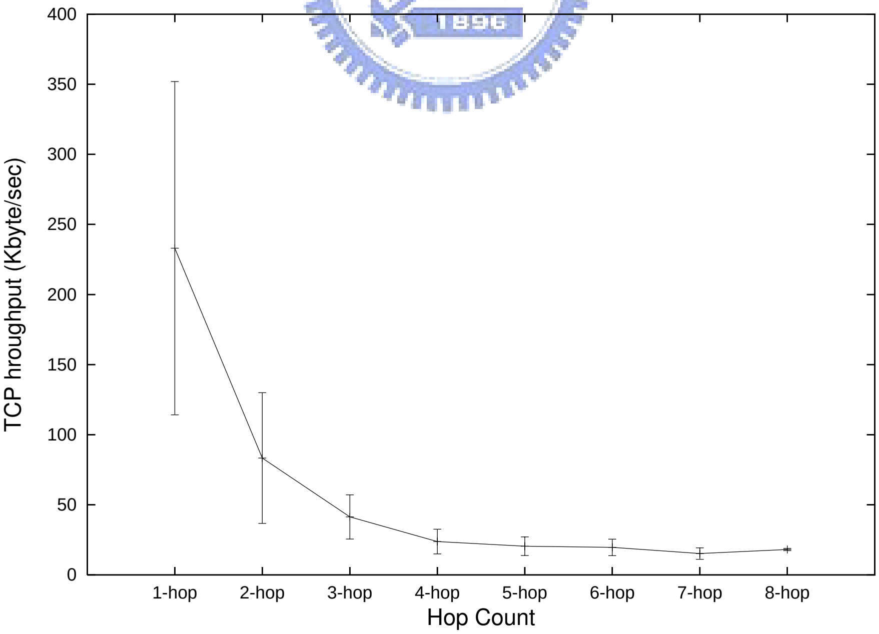


Figure 5.13: Client-to-client TCP performance

Chapter 6

Scalability Issues

In this chapter, we report the simulation speed and memory usage of our WiMAX Mesh simulation modules built on top of the NCTUns network simulator. NCTUns uses a discrete-event simulation method to advance its virtual clock. The more events it needs to process, the slower its simulation speed will be. Therefore, the number of events processed are also reported. The used machine for scalability testing is a desktop computer equipped with a 3.0 GHz Pentium processor and 2.0 GB RAM.

6.1 Number of Connections

A random network topology consisting of 50 nodes is used in all simulations performed in this section. We adopt a VoIP-like traffic pattern: In each simulation, there are a number of pairs of UDP constant-bit-rate connections whose source and destination are randomly generated. For comparison, a simulation case without any connections is also performed. The simulation time is 400 seconds and the traffic generators start sending data at 150 seconds. Each simulation is run for 10 times.

Assume a G.711 codec which takes a sample every 20 ms is used. The G.711 transmits 64000 bits per second. In other words, a sample consisting of 160 bytes is transmitted every 20 ms. The header overhead imposed by the Real-time Transport Protocol (RTP) is 12 bytes. Therefore, the CBR of each UDP connection is 172

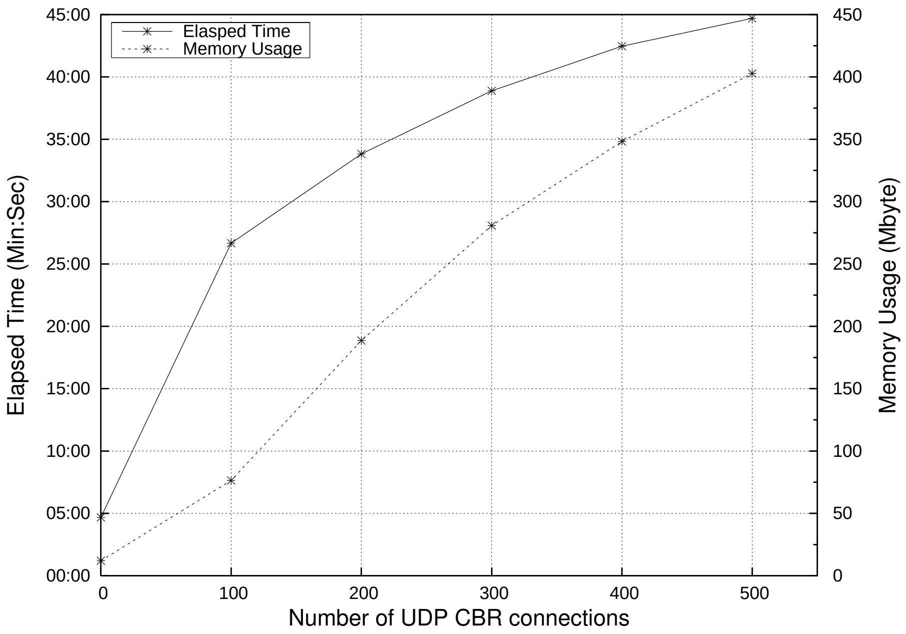


Figure 6.1: Number of UDP CBR connections vs. elapsed time and memory usage

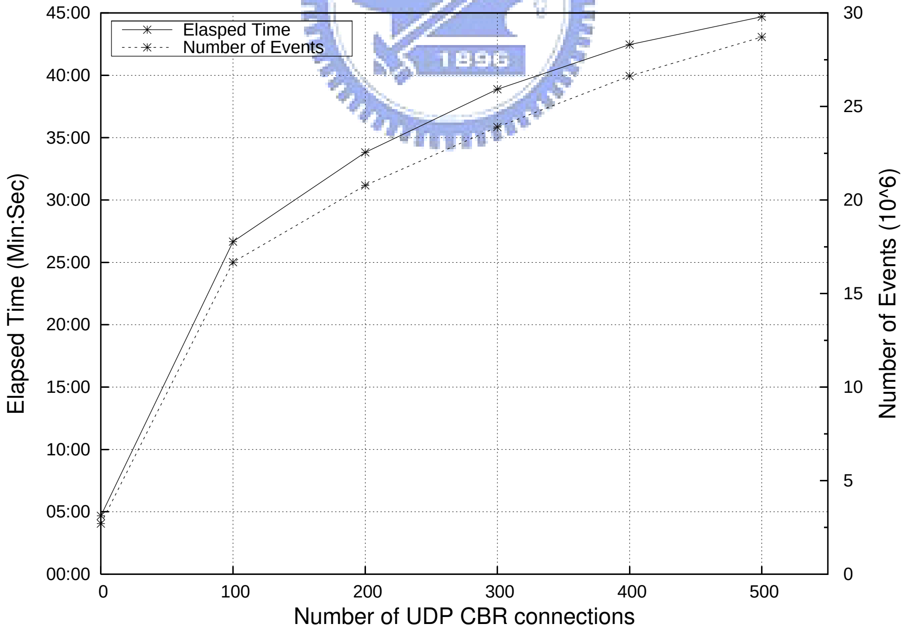


Figure 6.2: Number of UDP CBR connections vs. elapsed time and number of events

bytes per 20 ms.

The simulation results are shown in Fig. 6.1 and Fig. 6.2. It can be seen that only a moderate amount of memory resources are used. As expected, the execution time is proportional to the number of events processed by the simulation engine of the NCTUns network simulator.

6.2 Channel Model and Channel Coding

When channel model and channel coding are enabled, many byte-level and bit-level operations are performed by the randomizer/derandomizer, the RS encoder/decoder, the convolutional encoder/decoder, and the interleaver/deinterleaver. This may downgrade the simulator performance a lot. In this section, we observe the overhead caused by these operations.

We use the same network topology and traffic pattern as described in Section 6.1. For comparison, a simulation case without any connections is also performed. The simulation time is 150 seconds and the traffic generators start sending data at 100 seconds. Each simulation is run for 10 times.

The simulation results are shown in Fig. 6.3 and Fig. 6.4. We can see that the simulator performance is almost unacceptable. A possible solution to this problem will be mentioned in Chapter 7.

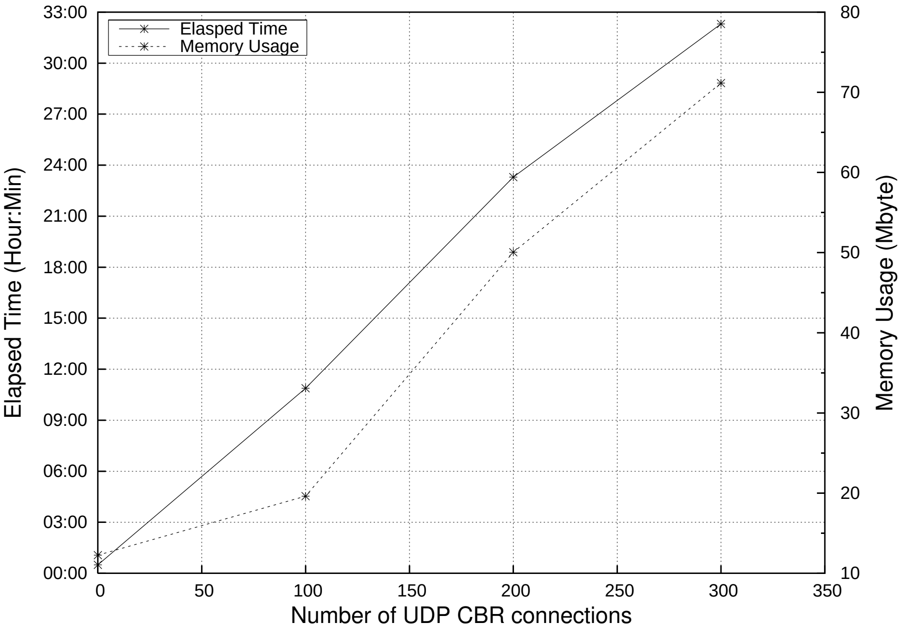


Figure 6.3: Number of UDP CBR connections vs. elapsed time and memory usage (under a more realistic wireless channel)

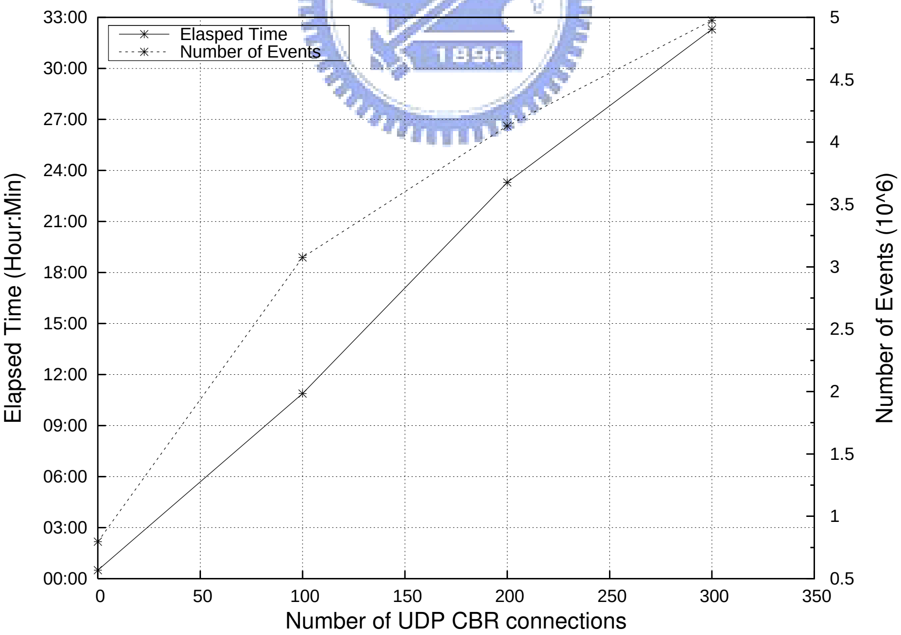


Figure 6.4: Number of UDP CBR connections vs. elapsed time and number of events (under a more realistic wireless channel)

Chapter 7

Future Work

- Automatic Repeat Request (ARQ)

  TCP performs poorly in error-prone wireless channels because it mistakenly believes that packet losses are due to network congestion. ARQ capability can be implemented in the MAC layer to reduce the packet loss rate experienced by TCP.

- Centralized Scheduling

  The distributed scheduling increases the level of spatial reuse in the mesh network. The centralized scheduling, on the other hand, allocates network resources in a more efficient way, since the Mesh BS has the network topology information such as the signal quality on each link. To better utilize network resources, combining these two approaches is a must.

- Channel Model and Channel Coding

  As shown in Chapter 6, the simulator performance is unacceptable when the feature of channel model and channel coding are enabled. A table-lookup approach is needed to improve the simulator performance while maintaining the correctness of the simulation. For example, the packet error ratio (PER) can be stored in a pre-computed table. This PER can then be used to determine whether or not an received packet gets dropped.

- Routing Protocol

  The Mesh mode defined in the standard does not provide broadcast function for its upper layer protocol. However, typical wireless routing protocols are built on top of this function. Thus, a mapping between broadcast and multi-unicast is needed. This mapping function can be implemented either in the CS layer or in an extra layer inserted between the MAC layer and the layer-3 routing protocol.

- Connection-Oriented MAC and QoS support

  Currently, the QoS in the Mesh mode is provisioned over links on a *message by message* basis. In this way, QoS over multihop can hardly be supported. Therefore, a connection-oriented MAC is needed for the peers in the WiMAX Mesh network to establish connections with QoS guarantee.

Chapter 8

Conclusion

The importance of WiMAX technology is increasing day by day. More and more researchers are studying critical issues of WiMAX networks. Therefore, a high-quality and general-purposed WiMAX network simulator is desired.

In this thesis, we describe the detailed design and implementation of our WiMAX Mesh simulation system over the NCTUns network simulator. Functionalities such as coordinated distributed scheduling, control message exchanging, and the network entry process are implemented and validated. We also evaluate the performance of WiMAX Mesh networks and discuss scalability issues.

To the best of our knowledge, this is the first public WiMAX simulation system developed over a general-purposed network simulator. Based on our work, researchers can further improve the simulation system or develop their own mechanisms on top of it.

Bibliography

[1] 802.16d 2004. Draft IEEE standard for local and metropolitan area networks - Part 16: Air interface for fixed broadband wireless access systems. May 2004.

[2] I. F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: A Survey. *Computer Networks*, 47(4):445–487, March 2005.

[3] 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 *ACM MobiHoc*, Urbana-Champaign, Illinois, USA, May 2005.

[4] V. Erceg, L. J. Greenstein, S. Y. Tjandra, S. R. Parkoff, A. Gupta, B. Kulic, A. A. Julius, and R. Bianchi. An empirically based path loss model for wireless channels in suburban environments. *IEEE Journal on Selected Areas in communications,*, 17(7):1205–1211, July 1999.

[5] V. Erceg, K. S. Hari, M. Smith, D. Baum, K. Sheikh, C. Tappenden, J. Costa, C. Bushue, A. Sarajedini, R. Schwartz, D. Branlund, T. Kaitz, and D. Trinkwon. Channel models for fixed wireless applications. July 2001.

[6] C. Hoymann. Analysis and performance evaluation of the OFDM-based metropolitan area network IEEE 802.16. *Computer Networks*, 49(3):341–363, March 2005.

[7] http://rscode.sourceforge.net/. RSCODE project.

[8] D. Kim and A. Ganz. Fair and efficient multihop scheduling algorithm for IEEE 802.16 BWA systems. In *IEEE BROADNET*, Boston, Massahusetts, USA, October 2005.

[9] K. N. Ramachandran, M. M. Buddhikot, G. Chandranmenon, S. Miller, E. M. Belding-Royer, and K. C. Almeroth. On the design and implementation of infrastructure mesh networks. In *IEEE WiMesh*, Santa Clara, CA, September 2005.

[10] S. Wang, C. Chou, C. Huang, C. Hwang, Z. Yang, C. Chiou, and C. Lin. The design and implementation of the NCTUns 1.0 network simulator. *Computer Networks*, 42(2):175–197, June 2003.

[11] H.-Y. Wei, S. Ganguly, R. Izmailov, and Z. J. Haas. Interference-aware IEEE 802.16 WiMax mesh networks. In *IEEE VTC*, Stockholm, Sweden, May 2005.