國立交通大學

電信工程學系

碩士論文

IEEE 802.16e 服務品質保證之
省電機制排程演算法

An Energy-Saving Scheduling Algorithm with QoS Guarantee for IEEE 802.16e Broadband Access Networks

研究生：蔡耀誼

指導教授：李程輝 教授

中華民國九十七年十月

IEEE 802.16e 服務品質保證之
省電機制排程演算法

An Energy-Saving Scheduling Algorithm with QoS Guarantee for IEEE 802.16e Broadband Access Networks

研 究 生：蔡耀誼	Student：Yaw-Yi Tsai
指導教授：李程輝	Advisor：Tsern-Huei Lee

國立交通大學
電信工程學系
碩士論文

A Thesis
Submitted to Department of Communication Engineering
College of Electrical and Computer Engineering
National Chiao Tung University
In partial Fulfillment of the Requirements
For the Degree of
Master
In

Communication Engineering

October 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年十月

IEEE 802.16e 服務品質保證之省電機制排程演算法

學生：蔡耀誼　　　　　　　　　　指導教授：李程輝 教授

國立交通大學電信工程學系碩士班

中文摘要

本篇論文即根據此睡眠機制，針對非請求的頻寬分配(UGS)的服務、固定位元率(CBR)，提出一種排程演算法，目的既要達到服務品質保證(QoS)，即不違反延遲界限(delay bound)，又要達到省電，以及頻寬使用最大化。

此排程演算法根據比率單調排程(Rate Monotonic Scheduling)而設計，每個系統內的工作站都具有靜態優先權(static priority)，當工作站達到可以接受服務的條件，才可以進行服務，萬一超過一個工作站符合服務條件，則基地台會挑選優先權最高的工作站進行服務。另外，針對這個排程演算法，我們討論其允入控制(admission control)，先探討系統內多個工作站的可排性(schedulability)，再推導下一個系統可以允許進入服務的新工作站，其最小的封包抵達時間間隔(inter-arrival time)為多少。最後並將此演算法與其他方法做省電效能與系統吞吐量的比較。

An Energy-Saving Scheduling Algorithm with QoS Guarantee for IEEE 802.16e Broadband Access Networks

Student: Yaw-Yi Tsai Advisor: Dr. Tsern-Huei Lee

Institute of Communication Engineering
National Chiao Tung University

Abstract

This thesis focuses on the scheduling problem for CBR traffic with delay constraint in IEEE 802.16e broadband access networks. Multiple MSSs are considered in our work. In order not to violate the delay constraint, and maximize both the energy efficiency and bandwidth utilization, we propose a scheduling algorithm based on rate monotonic scheduling for CBR traffic. In this scheduling algorithm, any connection would be served when it becomes eligible. The highest priority would be selected for service when there is more than one eligible connection. Moreover, we discuss the admission control for this algorithm which aims to obtain the minimum inter-arrival time of next acceptable connection. The numerical results show that our scheduling algorithm performs well on energy efficiency and bandwidth utilization.

誌謝

感謝指導教授—李程輝老師，兩年碩士班的訓練，從研讀學術期刊會議論文起，到遇上研究問題該有的嚴謹思考態度，以求有效的解決問題，老師的經驗傳承與精準的指導，確實導正了我很多思考面向。謝謝老師！

感謝我摯愛的雙親—蔡崇吉先生與陳玉琴女士，自小教導我正確的生活態度，告訴我待人處事務求誠懇踏實，眼界要寬，心地要亮，為人要正直，求學態度亦如是。感謝我最敬愛的奶奶，幼時的人格教育，與如今偶爾的叮嚀，我感念謹記在心，希望您的身體能一天一天好轉。

感謝郁文學長在這篇論文研究，花了很多的時間，一起討論這個研究課題，從他的身上，我看到了學術研究的熱誠。感謝景融學長對於我研究上的鼓勵，時常給予細心的解釋與幫助。謝謝兩位學長無私的幫忙。

感謝學長姐及同窗們—瑋哥、迺倫、博文、柏庚、建成、世弘、凱文、勁文、明鑫、錫堯、明智，一起努力打拼。感謝一群可愛的學弟們—松晏、鈞傑、家豪、俊德、佑信、敬堯、晨屹，有你們的實驗室很融洽。感謝小汪在寫論文的日子裡，幫我找回了對於籃球的衝勁與熱情。感謝 Ginny 在論文完成的後段幫了我很多忙。謝謝大家！

謹將此論文獻給所有我愛與愛我的人。

2008年10月 於風城交大

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over recent decades, the explosive growth of the Internet has resulted in a substantial increase in demand for high speed and ubiquitous Internet access. Broadband Wireless Access (BWA) has the potential to satisfy these demands. IEEE Standard 802.16-2004 (World Interoperability for Microwave Access - WiMAX) is a broadband wireless communication technology for last mile access, which aims to provide high data rate, low cost between Subscriber Stations (SSs) and Base Station (BS). It is expected to fill up the gap between high data rate in wireless Local Area Networks and low data rate in mobile cellular systems.

However, IEEE Standard 802.16e does not provide mobility for SSs. A new modified version, IEEE Standard 802.16e (Mobile World Interoperability for Microwave Access - Mobile WiMAX) was proposed for providing services between Mobile Subscriber Stations (MSSs) and BS.

Moreover, real-time services such as Voice over Internet Protocol (VoIP), Video on Demand (VOD) have become widely-used, popular Internet applications. In order not to violate the guaranteed packet delay and packet loss constraints of each connection, issues in

Quality of Service (QoS) have become quite important for the design of Medium Access Control (MAC) Protocol, the original IEEE standard 802.16e MAC protocol does not particularly address this aspect though. On the other hand, for continuously providing convergence and seamless services to users, we have to explore energy management of MSSs in IEEE 802.16e systems.

In this thesis, we focus on the scheduling problem of the real-time services between BS and MSSs in IEEE 802.16e broadband wireless access OFDM networks. While serving the MSSs, the most effective way to reduce power consumption is to make them enter sleep mode periodically. In the sleep mode, MSSs would turn off the transceiver to save power. The BS which provides services to those MSSs has to buffer the downlink traffic for those MSSs who are in the sleep mode. The MSSs in sleep mode would wake up in the listening interval periodically to receive the buffered data from the BS. After that, they return to the sleep mode again. However, each connection of a MSS in the IEEE 802.16e networks has its own delay requirement, which implies the buffered data must be sent or received before its delay constraint expires.

In order to support a wide variety of applications, IEEE 802.16e defines five scheduling services: Unsolicited grant services (UGS), Real-time polling services (rtPS), Non-real-time polling service (nrtPS), Best-effort (BE) service, Extended real-time variable rate (ert-VR) service. In this thesis, we only consider the UGS traffic. The UGS traffic is designed to support fixed-size data packets at CBR, such as T1/E1 emulation and VoIP without silence suppression. Therefore, our objective is to find a scheduler for UGS to satisfy the delay constraints, maximize power saving, and maximize the bandwidth utilization.

We propose a scheduling algorithm and admission control for CBR traffic in IEEE

802.16e broadband wireless OFDM networks. Here, we only consider the downlink traffic. The second power saving class in IEEE 802.16e standard is used. In our approaches, we assume that the delay bound of each connection is a multiple of its inter-arrival time, which is described as $D = \Omega T$. Based on the Rate-Monotonic Scheduling, we determine a set of scheduling policies that a connection would be served when it accumulates the most packets in the queue so that it will save the most energy for a MSS. In order not to violate the delay constraint, a connection should be served before the head of line goes out of date. Furthermore, for a connection which is in service at a specific instant, the BS can calculate the next instant that it will be served. The BS should notice the MSSs that how long they could sleep for and when they should wake up. Also, for those existing connections, we can compute in advance the minimum inter-arrival time of the next acceptable connection by capitalizing on the empty-minus-backlog curve.

The remainder of this paper is organized as follows. In Chapter 2, the MAC mechanism in IEEE 802.16 and the enhanced one in IEEE 802.16e are described. A survey of related work about scheduling algorithm and admission control for wireless network is given in Chapter 3. We introduce the Rate-Monotonic Scheduling in this chapter as well. Chapter 4 and Chapter 5 respectively present our proposed algorithm (including the system model) and admission control based for IEEE 802.16e OFDM networks. Chapter 6 shows the performance evaluation with the numerical results of our proposed algorithm. Finally, we draw our conclusions and future work in Chapter 7.

Chapter 2

Backgrounds

IEEE Standard 802.16 (WiMAX) was proposed to solve the problem of the wireless access network for last-mile. It would provide the convenience of the connection on the Internet in rural areas or in damaged places which suffered from the unpredictable disasters. In IEEE 802.16e, there are two operation modes, Point-to-multipoint, and mesh. PMP is the most popular way to access WiMAX.

Point-to-point applications include interbuilding connectivity within a campus and microwave backhaul. Point-to-multipoint applications include (1) broadband for residential, small office/home office (SOHO), and small- to medium-enterprise (SME) markets, (2) T1 or fractional T1-like services to businesses, and (3) wireless backhaul for Wi-Fi hotspots. Figure 2.1 illustrates the various point-to-multipoint applications.

Consumer and small-business broadband applications are one kind of the largest applications of WiMAX in the near future which is likely to be the broadband access for residential, SOHO, and SME markets. Broadband services provided using fixed WiMAX could include high-speed Internet access, telephony services using voice over IP, and a host of other Internet-based applications. T1 emulation for business is the other major opportunity for fixed WiMAX in developed markets which is as a solution for competitive T1/E1, fractional T1/E1, or higher-speed services for the business market. Given that only a small fraction of commercial buildings worldwide have access to fiber, there is a clear need for alternative high-bandwidth solutions for enterprise customers. Backhaul for Wi-Fi hotspots is another interesting opportunity for WiMAX in the developed world which is the potential to serve as the backhaul connection to the burgeoning Wi-Fi hotspots market.

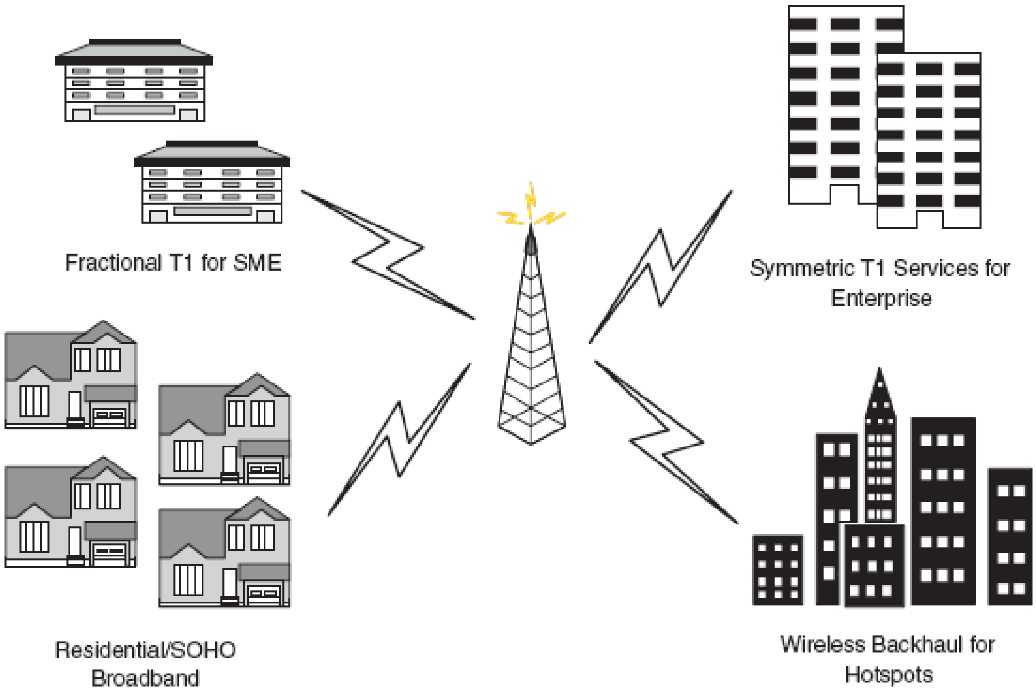


Figure 2.1 Point-to-multipoint WiMAX applications

The first version of IEEE standard 802.16 does not address the mobility. It only provides the services for fixed SS. However, as the growing needs for real-time services of wireless devices, those fixed things can not satisfy the customers anymore. Therefore, a modified standard of WiMAX, IEEE Standard 802.16e has been proposed by WiMAX Forum in 2005.

In IEEE 802.16e, it not only adds in the mobility of the system but also introduce the sleep mode into the MAC layer protocol. Three power saving classes are described in IEEE standard 802.16e.

2.1 Scheduling Services in IEEE Standard 802.16【5】

To support a wide variety of applications, WiMAX defines five scheduling services (Table 2.1) that should be supported by the base station MAC scheduler for data transport over a connection:

1. **Unsolicited grant services (UGS):** This is designed to support fixed-size data packets at a constant bit rate (CBR). Examples of applications which may use this service are T1/E1 emulation and VoIP without silence suppression.

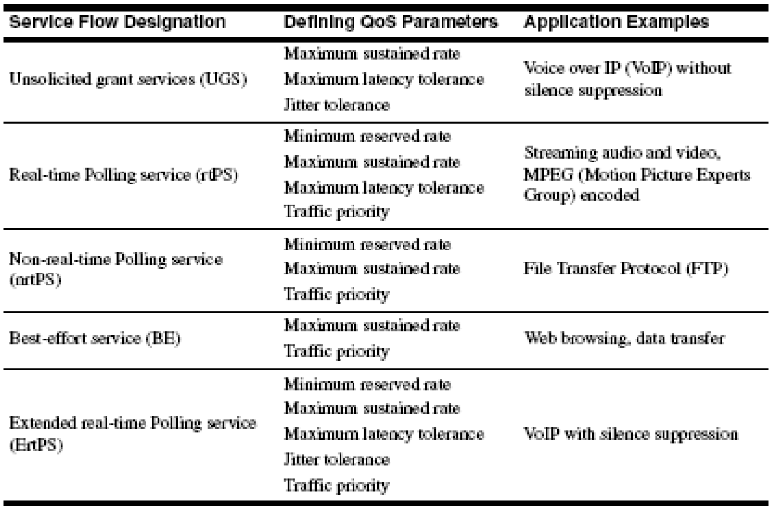
2. **Real-time polling services (rtPS):** This service is designed to support real-time service flows, such as MPEG video, that generate variable-size data packets on a periodic basis.

3. **Non-real-time polling service (nrtPS):** This service is designed to support delay-tolerant data streams, such as an FTP, that require variable-size data grants at a minimum guaranteed rate.

4. **Best-effort (BE) service:** This service is designed to support data streams, such as Web browsing, that do not require a minimum service-level guarantee.

5. **Extended real-time variable rate (ERT-VR) service:** This service (only in IEEE

802.16-2005) is designed to support real-time applications, such as VoIP with silence suppression, that have variable data rates but require guaranteed data rate and delay. This is also referred to as extended real-time polling service (ErtPS).



Service Flow Designation	Defining QoS Parameters	Application Examples
Unsolicited grant services (UGS)	Maximum sustained rate Maximum latency tolerance Jitter tolerance	Voice over IP (VoIP) without silence suppression
Real-time Polling service (rtPS)	Minimum reserved rate Maximum sustained rate Maximum latency tolerance Traffic priority	Streaming audio and video, MPEG (Motion Picture Experts Group) encoded
Non-real-time Polling service (nrtPS)	Minimum reserved rate Maximum sustained rate Traffic priority	File Transfer Protocol (FTP)
Best-effort service (BE)	Maximum sustained rate Traffic priority	Web browsing, data transfer
Extended real-time Polling service (ErtPS)	Minimum reserved rate Maximum sustained rate Maximum latency tolerance Jitter tolerance Traffic priority	VoIP with silence suppression

Table 2.1 Service Flows Supported in WiMAX

2.2 Sleep Mode for mobility-support MS【4】

Sleep mode is a state in which an MS conducts pre-negotiated periods of absence from the Serving BS air interface. These periods are characterized by the unavailability of the MS, as observed from the Serving BS, to downlink (DL) or uplink (UL) traffic. Sleep mode is intended to minimize MS power usage and decrease usage of Serving BS air interface resources. Implementation of sleep mode is optional for the MS and mandatory for the BS.

For each involved MS, the BS keeps one or several contexts, each one related to a certain power saving class. A power saving class is a group of connections that have common demand properties. For example, all BE and NRT-VR connections may be marked as belonging to a single class while two UGS connections may belong to two different classes in case they have different intervals between consequent allocations. A power saving class may be repeatedly activated and deactivated. Activation of certain power saving class means starting sleep/listening windows sequence associated with this class. Algorithm of choosing power saving class type for certain connections is outside of the scope of the standard.

There are three types of power saving classes, which differ by their parameter sets, procedures of activation/deactivation, and policies of MS availability for data transmission. Figure 2.2 describes example of behaviors of MS with two Power Saving Classes: Class A contains several connections of BE and NRT-VR type, Class B contains a single connection of UGS type. Then for Class A the BS allocates sequence of listening window of constant size and doubling sleep window. For Class B the BS allocates sequence of listening window of constant size and sleep window of constant size. The MS is considered unavailable (and may power down) within windows of unavailability, which are intersections of sleep windows of A and B.

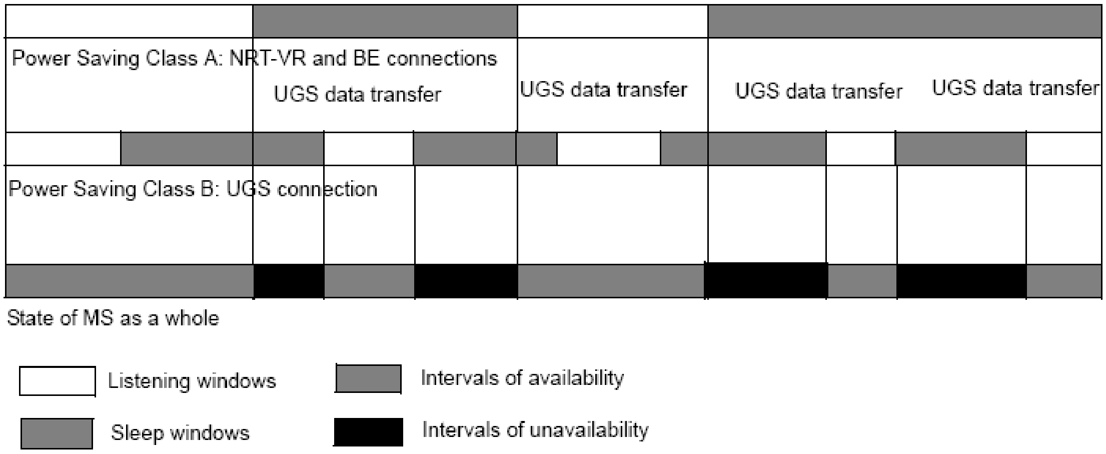


Figure 2.2 Example of sleep mode operations with two power saving classes

The three power saving classes would be described as follows. In the first type power saving class which is suitable for web browsing or data access services, a connection sleeps for a specific period of time, and then wake up for listening that whether there is any traffic for it or not. If there is no packet sent or received, a MSS doubles its next sleep period. The length of the listening periods is fixed. In the second power saving class, the rules are the same as the first power saving class, except that the length of the sleeping and listening periods are fixed. This class fits the real-time applications such as VoIP or video stream services. The third power saving class, the length of the sleep period is predefined. The MSS sleeps for a defined period of time and then goes back to the normal mode. It is recommended for multicast connections as well as for management operations, such as periodic ranging, DSx operations. Figure 2.3 shows the three power saving classes

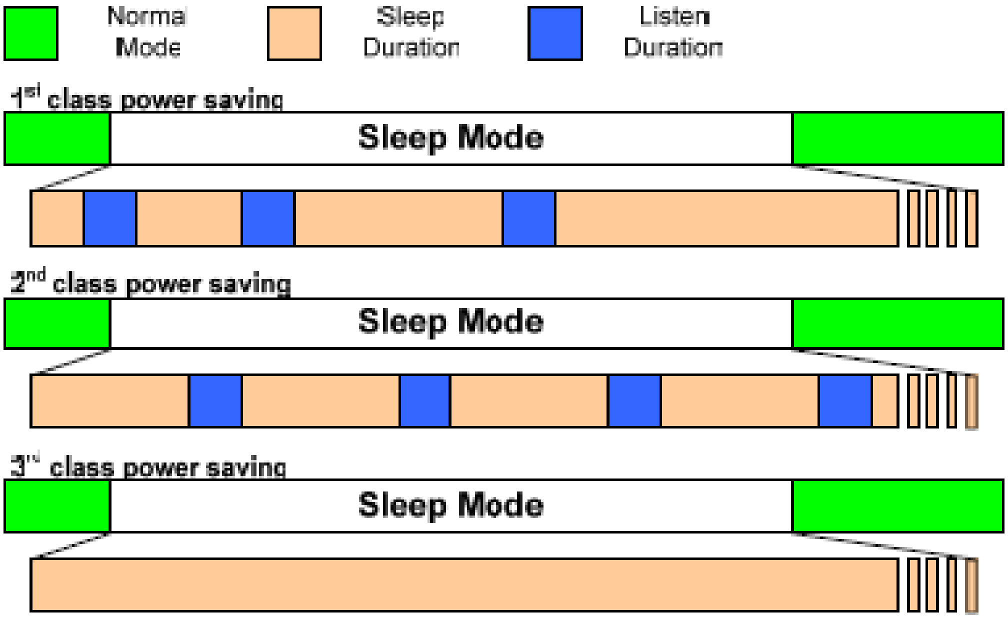


Figure 2.3 The three power saving classes

2.3 OFDM Frame Structure【4】

In this thesis, due to that we discuss about the IEEE OFDM network, it is necessary to introduce the OFDM frame structure. Figure 2.4 shows an OFDMA and OFDM frame when operating in TDD mode. The frame is divided into two subframes: a downlink frame followed by an uplink frame after a small guard interval. The downlink-to-uplink-subframe ratio may be varied from 3:1 to 1:1 to support different traffic profiles.

WiMAX also supports frequency division duplexing, in which case the frame structure is the same except that both downlink and uplink are transmitted simultaneously over different carriers. Some of the current fixed WiMAX systems use FDD. Most WiMAX deployments, however, are likely to be in TDD mode because of its advantages. TDD allows for a more flexible sharing of bandwidth between uplink and downlink, does not require paired spectrum, has a reciprocal channel that can be exploited for spatial processing, and has a simpler transceiver design. The downside of TDD is the need for synchronization across multiple base stations to ensure interference-free coexistence. Paired band regulations, however, may force some operators to deploy WiMAX in FDD mode.

As shown in Figure 2.4, the downlink subframe begins with a downlink preamble that is used for physical-layer procedures, such as time and frequency synchronization and initial channel estimation. The downlink preamble is followed by a frame control header (FCH), which provides frame configuration information, such as the MAP message length, the modulation and coding scheme, and the usable subcarriers. Multiple users are allocated data regions within the frame, and these allocations are specified in the uplink and downlink MAP messages (DL-MAP and UL-MAP) that are broadcast following the FCH in the downlink subframe. MAP messages include the burst profile for each user, which defines the modulation and coding scheme used in that link. Since MAP contains critical information that needs to reach all users, it is often sent over a very reliable link, such as BPSK with rate 1/2 coding and repetition coding. Although the MAP messages are an elegant way for the base station to inform the various users of its allocations and burst profiles on a per-frame basis, it could form a significant overhead, particularly when there are a large number of users with small packets (e.g., VoIP) for which allocations need to be specified. To mitigate the overhead concern, mobile WiMAX systems can optionally use multiple sub-MAP messages where the dedicated control messages to different users are transmitted at higher rates, based on their individual SINR conditions. The broadcast MAP messages may also optionally be compressed for additional efficiency.

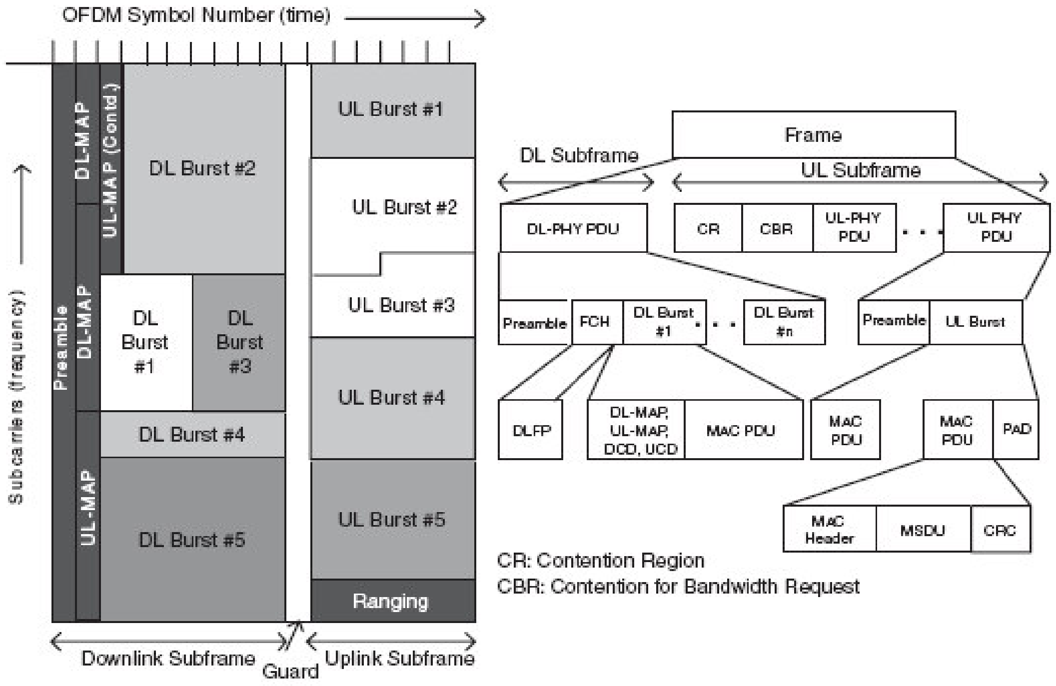


Figure 2.4 A sample TDD frame structure for Mobile WiMAX

Chapter 3

Related Works

There are many researches about the topics on Quality of Service (QoS) and energy saving in IEEE 802.16e. Part of the researches address on the scheduling problem. Though the IEEE standard 802.16e introduces the details of WiMAX, it does not address on the scheduling part. The challenge is that in order not to violate the QoS constraint and to reduce the most energy consumption, a scheduling algorithm has to be proposed for the traffic in the system. The following paragraphs will present some scheduling approaches for IEEE 802.16e, and a renowned scheduling algorithm called rate-monotonic scheduling which is for CPU processing.

3.1 Scheduling Approaches: MMPS【3】

Three scheduling approaches, MMPS, MMPS-FC, and MMPS-BF are proposed in [3] by Huang. In [3], the length of a common sleeping cycle is determined by the minimum of the sleeping cycles of all MSSs. It does reduce the energy consumption and does not violate the delay constraint of all the MSSs. However, for some MSSs whose sleeping cycles are shorter than the common cycle, it would save more energy if they maintain their own sleeping cycles.

They consider the second type of power saving class in IEEE 802.16e wireless network. The downlink CBR traffic is also considered. Suppose there are multiple MSSs in the system, and each of them has at least one UGS connection. They aim to find a scheduler to achieve the delay requirement of each connection and maximize the power saving. They would like to maximize the bandwidth utilization as well. Their proposed scheduling approaches would be described as follows.

Step1: Find a common fixed sleeping cycle for all MSSs to simplify their scheduling. As shown in figure 3.1, suppose there is a *MSS* which has three connections. Each of the connections has its own delay constraint, φ (frames). In order not violate each delay constraint. The sleeping cycle, T_{s_i}, for each $MSS\ S_i$ can be determined by

$$T_{s\_i} = \min\{\varphi_1, \varphi_2, \varphi_3, ..., \varphi_m\} \tag{3.1}$$

For all *MSSs*, the common fixed sleeping cycle, T_{com} (in units of OFDM frames), is determined as the minimum sleeping cycle over all serviced MSSs.

$$T_{com} = \min\{T_{s\_1}, T_{s\_2}, T_{s\_3}, ..., T_{s\_n}\} \tag{3.2}$$

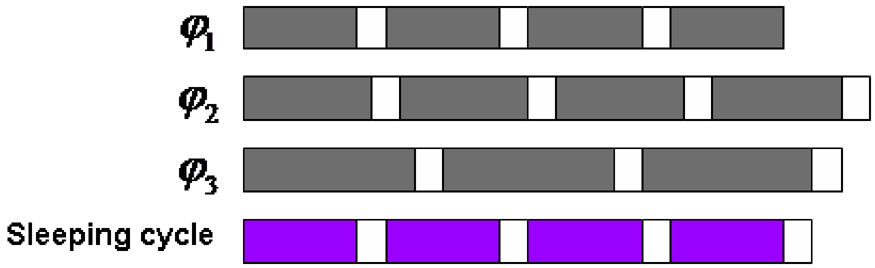


Figure 3.1 The sleeping cycle of a connection

Step 2: Allocate OFDM frames (bandwidth), denoted as ω, for each *MSS* during a common sleeping cycle T_{com}.

$$\omega_i = \left\lceil \frac{\tau_i \times T_{com}}{\Omega} \right\rceil \tag{3.3}$$

In the equation, τ_i is the total traffic generation rate of a MSS S_i and Ω is the capacity of an OFDM frame.

Step 3: Determine the number of MSSs, denoted as δ, which can be served by a BS. As shown in (3.4), the number of admitted MSSs must be less than or equal to the maximum number of MSSs where their allocated OFDM frames are smaller than the total bandwidth that can be supported by BS during T_{com}.

$$\delta = \arg\max_{j} \{ \sum_{i=0}^{j} \omega_i < T_{com} \} \tag{3.4}$$

Step 4: Schedule those admitted MSSs in round-robin rules. In order to make their allocated OFDM frames overlap free, the BS should delay the starting times of some traffic streams from some MSSs. Figure 3.2 shows the example of MMPS scheduling.

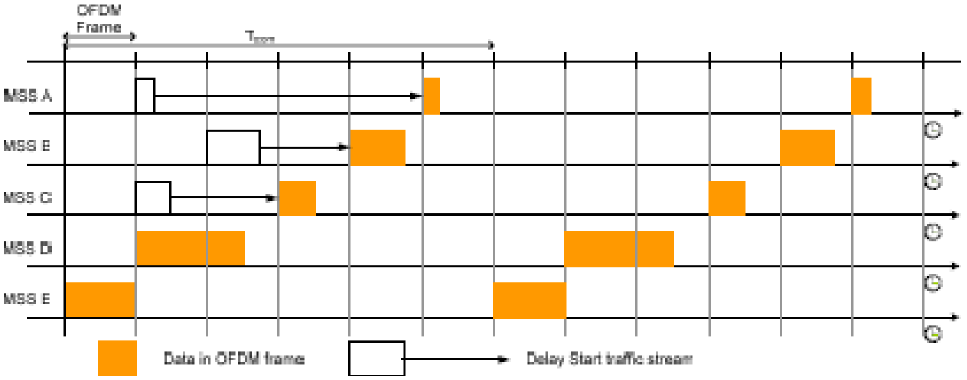


Figure 3.2 Example of MMPS scheduling

However, MMPS does not work well on bandwidth utilization. Fragment OFDM frames would happen in this proposed approach. A modified version of step 4 of MMPS is proposed to solve this problem. Equation (3.5) shows the modified step. Whenever a fragment OFDM frame is created by scheduling the packets for a new MSS, the BS selects the next MSS, S_x, whose required bandwidth is maximal but can fit into the remaining bandwidth of the fragment OFDM frame left by the previously scheduled MSS.

$$S_x = \max\left\{ frag(S_i) \right\} \text{ and } \quad frag(S_i) < \theta \tag{3.5}$$

Figure 3.3 shows the example of MMPS-FC.

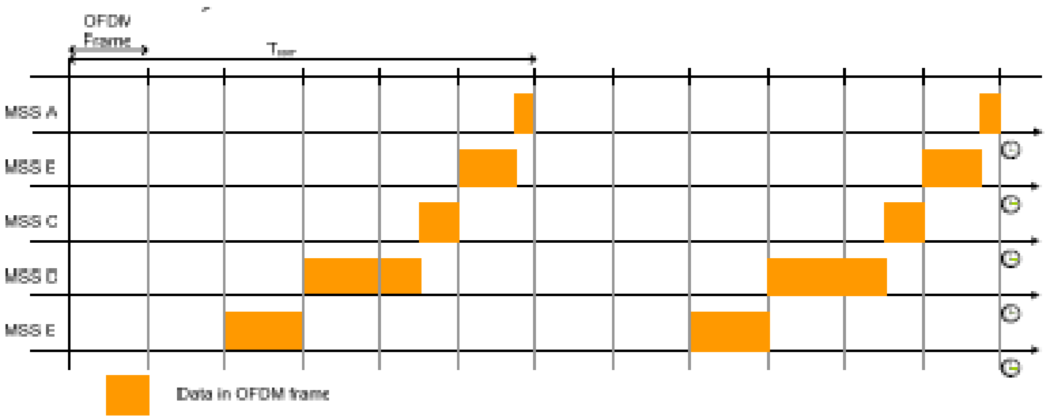


Figure 3.3 Example of MMPS with fragment collection

However, it still wastes some bandwidth if the fragment OFDM frame can not be completely fulfilled by the next MSS. In third approach, MMPS-BF, it breaks the boundary constraint. The scheduling can allocate bandwidth sharing across the boundary of two OFDM frames. Thus, no fragments are left. Figure 3.4 shows this example.

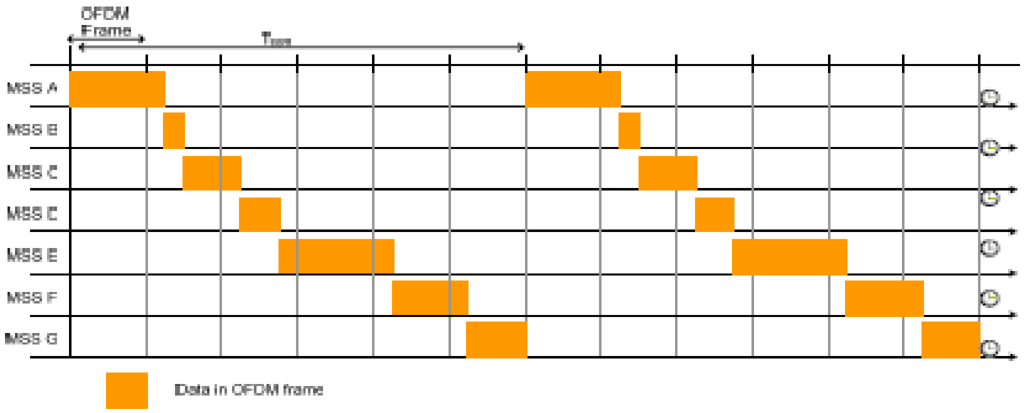


Figure 4. A MMPS example with boundary free.

Figure 3.4 MMPS example with boundary free

3.2 Rate-Monotonic Scheduling (RMS)【1】

In computer science, rate-monotonic scheduling is a scheduling algorithm used in real-time operating systems with a static-priority scheduling class. These operating systems are generally preemptive and have deterministic guarantees with regard to response times. Rate monotonic analysis is used in conjunction with those systems to provide scheduling guarantees for a particular application.

RMS is designed for the CPU processing in a hard-real-time environment. There some 5 assumptions.

(1) The requests for all tasks for which hard deadlines exist are periodic, with constant interval between requests.

(2) Deadlines consist of run-ability constraints only—i.e. each task must be completed before the next request for it occurs.

(3) The tasks are independent in that requests for a certain task do not depend on the initiation or the completion of requests for other tasks.

(4) Run-time for each task is constant for that task and does not vary with time. Run-time here refers to the time which is taken by a processor to execute the task without interruption.

(5) Any non-periodic tasks in the system are special; they are initialization or failure-recovery routines; they displace periodic tasks while they themselves are being run, and do not themselves have hard, critical deadlines.

The priorities are set up according to the request rate. The request rate of a task is defined to be the reciprocal of its request period. The higher the request rate is, the higher the priority is. The highest-priority ready process is selected for execution at a specific constant. Figure 3.5 shows an example of rate monotonic scheduling. At the beginning, two processes are ready for transmission. Process 1 gets the higher priority than process 2 due to its higher request rate, so that it would be served first.

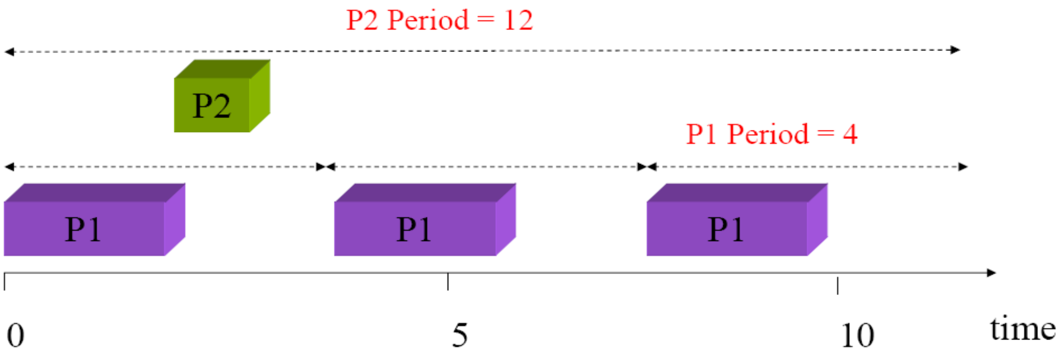


Figure 3.5 Example of rate monotonic scheduling

Chapter 4

Our Proposed Scheduling Algorithm

4.1 System model

This section gives the whole environment assumptions and the definition of a state that we used on our scheduling algorithm. Suppose that each MSS in a network has one downlink connection. Each connection provides UGS traffic, which generates fixed size data packets on a periodic basis. Besides, each connection of a MSS has a specific delay requirement in which the packets need to be delivered before its expiry time. All the sleeping and listening periods are in units of OFDM frames. Suppose that the following things are known: D_i, the delay bound of connection i; T_i, the inter-arrival time of connection i; m, numbers of MSSs; Ω, the capacity of an OFDM frame in the system, which implies that there are Ω packets that can be accommodated in a frame. In order to make the system simple, we assume that $D_i = \Omega T_i$. We consider the second type of power saving class in IEEE 802.16e wireless networks. Moreover, suppose that the BS knows that all the queue states of connections anytime. We define a state as follows.

We define a_i as the age the head of line in the queue of connection i (MSS i). Let

$a_i = 0$ at the time that the head of line arrives. It is clearly to see that $a_i \le 0$ when the queue is empty. To be more precise, the value would be represented as $-T_i + 1 \le a_i \le D_i - 1$. On the hand, in order not to violate the delay constraint, the value would be represented as $0 \le a_i \le D_i - 1$. Figure 4.1 is an example. Suppose that there is a connection whose inter-arrival time equals to 2 and delay bound equals to 4. Obviously, Ω is 2 in this case. The value a_i is set to be zero when the first packet arrives, and it increases progressively until it is served. As in the example, the packets are in service after $a_i = 2$, and then a_i becomes -1 due to that the MSS needs to wait for a frame for its next coming packet.

Let $\{a_i\}$ be the states of all queues of MSSs at a certain instant, where $i = 1 \sim m$.

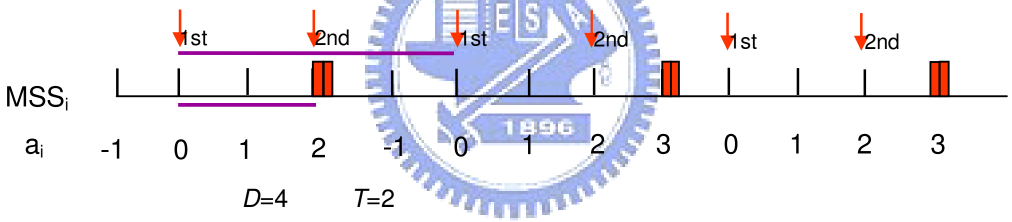


Figure 4.1 The example of a_i

4.2 Proposed Algorithm

In this thesis, our main goal is to propose an energy-efficient scheduling algorithm for CBR traffic in IEEE 802.16e OFDM wireless networks. With the assumption $D_i = \Omega T_i$, it is easy to see that the BS could accumulate more than one packets for a connection, and deliver these packets at a time instead of delivering one packet per time. Among these conditions, we can observe that, for a MSS, the most packets in the queue would be Ω. From the point of view of power saving, a MSS could only wake up and receive them when the queue is full of Ω packets, otherwise it enter the sleep mode. However, it decreases the system throughput if we let the connection only receive Ω packets at a time. We use the state mentioned in Section 2 to represent when a connection is ready to be served.

We call that a connection becomes eligible when it is ready to be served. A connection would request for the service from the BS when it becomes eligible. To unify the connections in the system, we separate the services into several conditions, i.e. $\Omega, \Omega-1, \ldots, \Omega-k, \ldots, 1$. Each condition has its eligible value which can be represented as a_i in Table 1. Take the Ω case for example, since a_i increases progressively and a connection becomes eligible when it accumulates Ω packets in the queue. The eligible value a_i for this case is from $D_i - T_i$ to $D_i - 1$. Here in the paper, we only consider the Ω case.

Eligible conditions	Eligible values
Ω	$D_i - 1 \geqq a_i \geqq D_i - T_i$
Ω, Ω-1	$D_i - 1 \geqq a_i \geqq D_i - 2T_i$
Ω, Ω-1, Ω-2	$D_i - 1 \geqq a_i \geqq D_i - 3T_i$
Ω, Ω-1, Ω-2, …, Ω-k	$D_i - 1 \geqq a_i \geqq D_i - (k+1)T_i$

Table 4.1 The eligible values of eligible conditions

4.2.1 Scheduling policies

Here in this thesis, we adopt the static priority scheduling. The followings are the scheduling policies.

Step 1: Prioritize all the connections in the system according to their inter-arrival time.

The smaller the inter-arrival time is, the higher the priority is. We can describe it as $T_1 \le T_2 \le \cdots \le T_m$. Connection 1 gets the highest priority, and connection m gets the lowest priority. Notice that the BS would randomly set up the priorities when $T_i = T_j$, for all $1 \le i, j \le m$, $i \ne j$.

Step 2: Set up the eligible values a_i of each MSS. Here, we set them as $D_i - T_i \le a_i \le D_i - 1$, i.e. the Ω case.

Step 3: A connection requests for service when it becomes eligible, which means it is ready to be served. The highest priority connection will be selected to be served when there is more than one eligible connection.

This scheduling algorithm can be described as follows. At the beginning of a frame, the BS picks an eligible connection *j* for service if and only if connection *j* has packets waiting for service and connection *i* does not for all $i < j$.

4.2.2 The waiting time of next service

Since the BS knows the inter-arrival time, the delay bound, and the state of each connection, it can calculate the waiting time of next service for each connection. That is to say, if connection i will be in service at a specific instant, the BS can notice connection i that how long it can sleep from this service and when it should wake up to be served next time. Figure 4.2 shows the waiting time of next service for connection i.

For any connection in the system, since we set up the Ω case, a connection can be served after the Ω^{th} packet arrives and before the next packet arrives. We call this interval as in eligible region. In figure 3, suppose that at a specific instant, connection i would be in service when its state is a_i. In the meantime, the state for connection k is a_k, where $k = 1 \sim i-1$. These connections are not eligible at this instant because if they are eligible, connection i would not be selected to be served at this instant.

We assume that l_i represents the interval between this service instant and the beginning of next eligible region. There are $i-1$ connections whose priorities are higher than connection i and they will be selected for service if they are eligible as well in the next eligible region of connection i. Therefore, the BS has to calculate how many connections among these $i-1$ connections will be in service in this region, and notice connection when it should wake up to receive data. Besides, for connection k $(k = 1 \sim i-1)$, suppose that l_k represents the interval between its state is a_k and the beginning of its next eligible region. It is clear to see, after the state a_k, the interval up to the beginning of the eligible region can be represented as $l_k + sD_k$, where D_k is the delay bound of connection k, and $s = 0, 1, 2, \ldots, etc$.

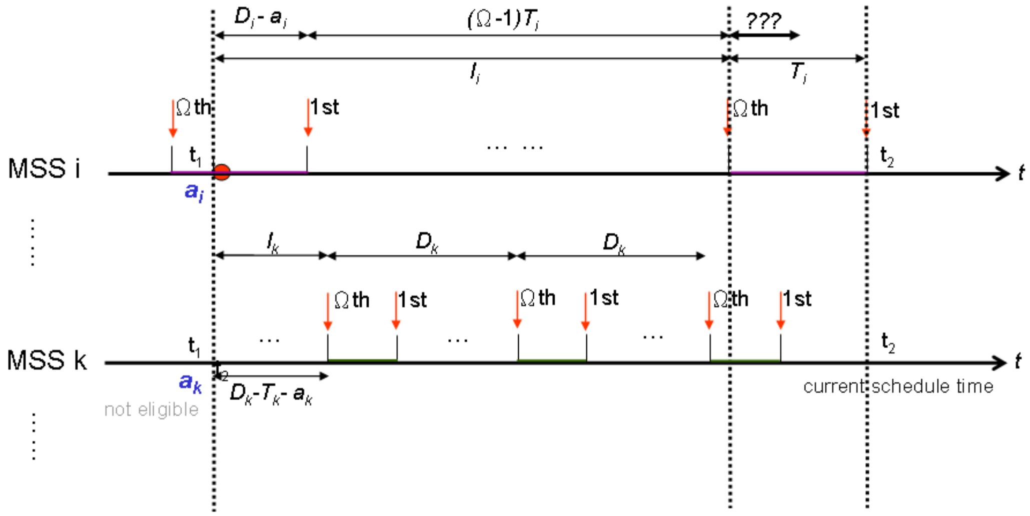


Figure 4.2 The waiting interval of the next service

For connection i, set $\{L\}=\{1,2,3,...,l_i+T_i-1\}$, which represents the intervals from t_1 to frames afterwards. The maximum element in this set is l_i+T_i-1. On the other hand, for connection k ($k=1\sim i-1$), set $\{L_k\}=\{l_k+sD_k\}$, for all $l_k+sD_k \le l_i+T_i-1$, and $s=0,1,2,...,etc$. Since these $i-1$ connections have higher priorities than connection i, we have to consider their schedules. While calculating these schedules, we let the BS start from connection 1 to connection $i-1$. The method would be described as Figure 4.3.

When a connection is chosen for service, the BS would set up its $\{L\}$ first. Then, the BS would find out all $\{L_k\}$. Starting from connection 1, the BS would check if there is any element in $\{L_k\}$ matches elements in $\{L\}$. Here, what we call matches means the same elements. If there is any match, it collects them in $\{L_k'\}$. If not, let the elements increases progressively, and check these values again. It would not stop this procedure till the value increases up to $l_k + sD_k + T$. After collecting all the elements of $\{L_k'\}$, we update $\{L\}$ as $\{L\} = \{L\} - \{L_k'\}$. Once it finishes the schedule of connection $i-1$, the waiting time of connection i would be the minimum in $\{L\}$ during the interval $(l_i, l_i + T_i)$.

After t_1, for any connection whose priority is lower than connection i, the BS already knew its service instant and next service instant if it does not exceed the current schedule time. If it exceeds the current schedule time, this connection should calculate the schedules again. On the other hand, for any connection whose priority is higher than connection i, the BS should definitely calculate again because it only records the schedules of lower priority connections.

The advantage of this method is that the BS would not have to calculate every time when there is any connection in service, i.e. it reduces complexity at some extent.

The algorithm for calculating the waiting time is shown at the end of this section.

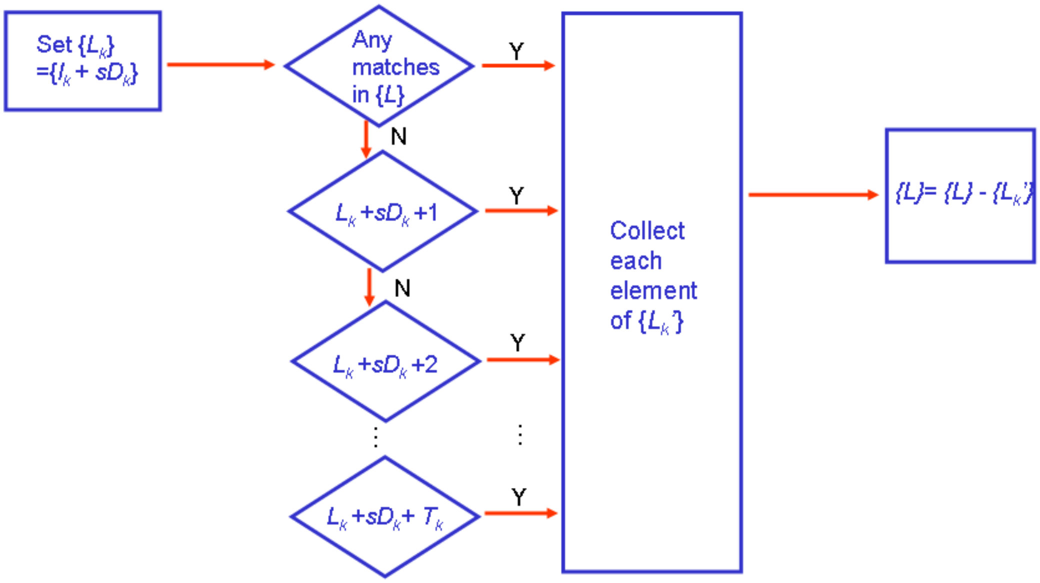


Figure 4.3 The flow chart of finding the waiting time of next service

Figure 4.4 is an example of the waiting time of next service. We can simply put it as follows. Suppose that there is a connection which is ready to be served, the BS should look ahead to the end of its next eligible region. If it exceeds the current schedule time, the BS should re-calculate and schedule the exceeding time interval. If not, the BS does not have to do anything due to it already know its next service time.

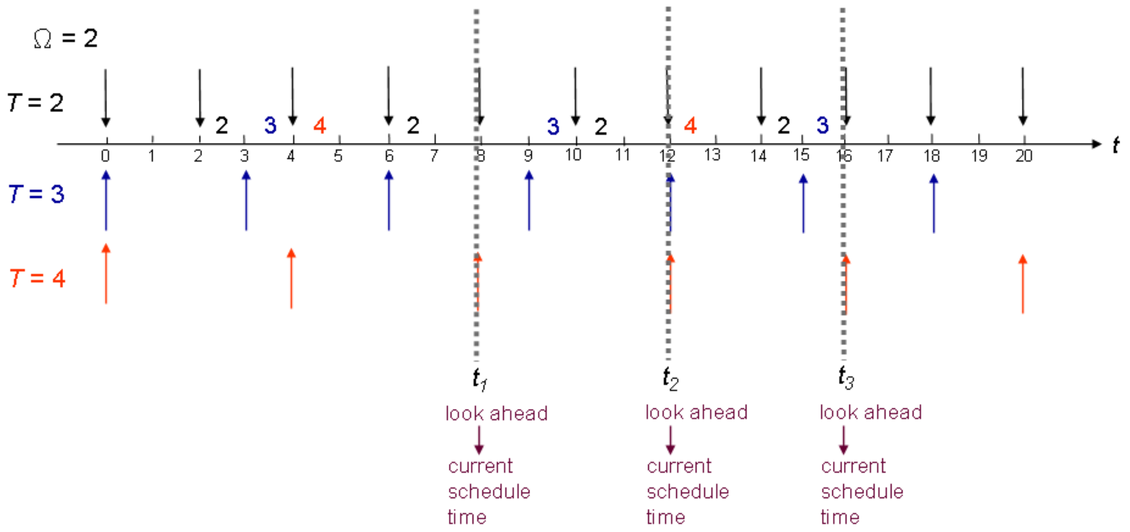


Figure 4.4 Example of the waiting time of next service

Scheduling algorithm of the waiting time of next service

```
While ( $Service \| StartTransmission$ ) do
    $i = priority\_service$
    $end\_of\_next\_eligible\_time = 2\Omega T_i - a_i$
    If ( $end\_of\_next\_eligible\_time > look\_ahead$ ) then
```

```
        $schedule\_time = end\_of\_next\_eligible\_time - look\_ahead$
        **For** ( $j = 1; j = j + 1 : j < i$ )
            **FrameAllocation** ( $j, schedule\_time$ )
        **End For**


**FrameAllocation** ( $j, schedule\_time$ )
$L(i) = \{1, 2, \ldots, schedule\_time\}$
$start\_of\_next\_eligible\_time = l_j$
    **For** ( $t = 0; t = t + 1; l_j + tD_j < schedule\_time$ )
    $s = l_j + tD_j$
    **For** ( $k = 1; k = k + 1; k \le schedule\_time$ )
        **If** ( $s = k$ ) **then**
            $L(i) = L(i) - k$
        **Else**
            **For** ( $w = 1; w = w + 1; w < T_j$ )
                $s = s + w$
                **If** ( $s = k$ ) **then**
                    $L(i) = L(i) - k$
```

Chapter 5

Admission Control

In this section, we would like to discuss the admission control of our proposed scheduling algorithm. As in [2], we discuss the schedulability test first. We have to derive a schedulable condition. And then we adopt the empty-minus-backlog curve to obtain the minimum inter-arrival time of the next acceptable connection. Besides, connection i is schedulable iff every packet of connection i is served within D_i frames after it arrives; and $\mathbf{D}_{\mathbf{1..m}}$ is schedulable if connection i is schedulable for all i . (where $\mathbf{D}_{\mathbf{1..m}} = [D_1, D_2, D_3, \ldots, D_m]$) represents these m connections with delay bounds)

5.1 Schedulability test

As mentioned above, a connection becomes eligible after the Ω^{th} packet arrives at the queue. This connection has to be served before the head of line goes out of date. Figure 5 shows the eligible region of a connection. ($D = \Omega T$) A connection is requested for service when it becomes eligible. In Figure 5.1, this connection is requested at the beginning of the eligible region. As in [1], we define the response time as the time span between the request and the end of the response to that request. For example, if a packet arrives at the beginning of frame *n* and is served in frame *k*, its response time is $k-n+1$.

Also, we define a critical instant as an instant at which a request for that connection will have the largest response time. We have the following theorem: A critical instant for any connection occurs whenever it is requested simultaneously with the requests for all higher priority connections. It can be proved in figure 6. Let connection 1, connection 2, … , and connection *m* denote a set of priority-ordered connections with connection *m* being the connection with the lowest priority. Consider a particular request for connection m that occurs at t_1. Suppose that between t_1 and t_1+T_m, the time at which the subsequent request of connection m occurs, requests for connection *i*, $i < m$ occur at t_2, $t_2 + 2T_1$, …, $t_2 + kT_1$ as illustrated in figure 6. Clearly, connection *i* will be in scheduled first after t_1, and each of them will accommodate an OFDM frame. Hence, for connection *m*, it would get the largest response time if it is requested at the time that connection *i* is requested as well. Repeating the argument for all connection *i*, $i = 2, …, m-1$, we prove that theorem that a critical instant for any connection occurs whenever the connection is requested simultaneously with requests for all higher priority connections.

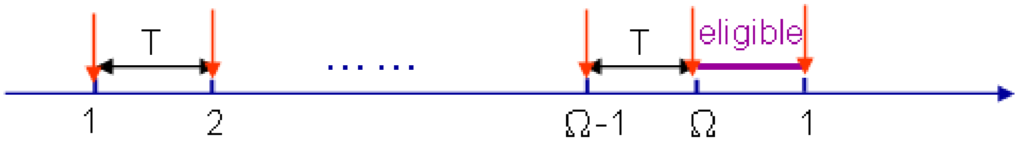


Figure 5.1 The eligible region of a connection

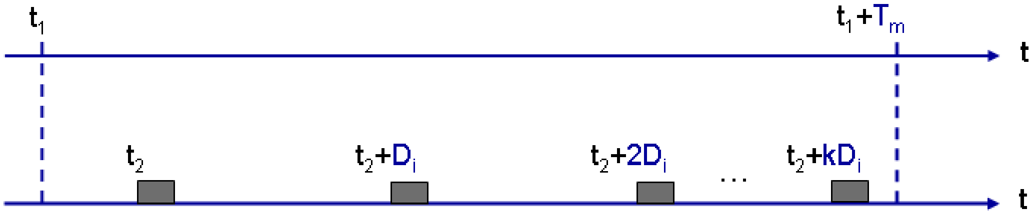


Figure 5.2 Services of connection 1 between requests for connection m

As a consequence, to study the schedulability problem of a set of m connections, we need only consider the situation where every connection is requested at the same time, i.e. all the connections just become eligible and ready to receive Ω packets. We only need to consider the situation that every connection generates the Ω^{th} packet at the beginning of the frame 1.

It is known in [2] that the schedulable condition for rate-monotonic scheduling is described as follows: Connection *k* is schedulable iff there exists an f_k such that $1 \le f_k \le T_k$, and $\sum_{j=1}^{k} \lceil f_k / D_j \rceil \le f_k, 1 \le f_k \le D_k$, where D_j represents the delay bound of connection *j*. This condition occurs if all the connection generates a packet at the beginning of frame 1.

We can modify this statement to meet our conditions: Connection *k* is schedulable iff there exists an f_k such that $1 \le f_k \le T_k$, and

$$\sum_{j=1}^{k} \left\lceil \frac{f_k}{\Omega T_j} \right\rceil \le f_k, 1 \le f_k \le T_k \tag{5.1}$$

The left side of the formula means the numbers of requests created by connection *j*, where $j = 1 \sim k$. If we can find an interval f_k and f_k is bigger than or equal to the number of requests up to f_k, we can say connection *k* is schedulable.

5.2 The empty-minus-backlog curve

As in [2], we define the empty-minus-backlog curve $H(f)$, $1 \le f \le C$, where C is the upper bound of the maximum inter-arrival time of any connection. The function $H(f)$ is as follows,

$$H(f) = f - \sum_{j=1}^{m} \left\lceil \frac{f}{\Omega T_j} \right\rceil \tag{5.2}$$

An example of $H(f)$ for $\mathbf{D_{1..3}} = [4, 4, 6]$ is shown in figure 5.3. It is clear to see that $\mathbf{T_{1..3}} = [2, 2, 3]$ due to Ω=2 in this case. ($\mathbf{T_{1..m}} = [T_1, T_2, T_3, \ldots, T_m]$ represents these *m* connections with inter-arrival time.) In this figure, $H(f)$ is drawn as a continuous curve for clarity and the value of *H*(*f*) is shown at the end of frame *f*. We let $H(0) = -m$ so that $-H(0) = m$ represents the number of requests waiting for service, i.e., backlog at the beginning of frame 1. Besides, on the x-axis, the number T_i is shown in frame *f* if connection *i* is served in that frame. In the function $H(f)$, $\sum_{j=1}^{m} \lceil f / \Omega T_j \rceil$ means the total number of requests up to frame *f* so that $H(f)$ represents the total number of frames minus the total number of requests up to frame *f*. Notice that the frame without an number on that frame is called an empty frame. It means that there is no connection would be served in the frame.

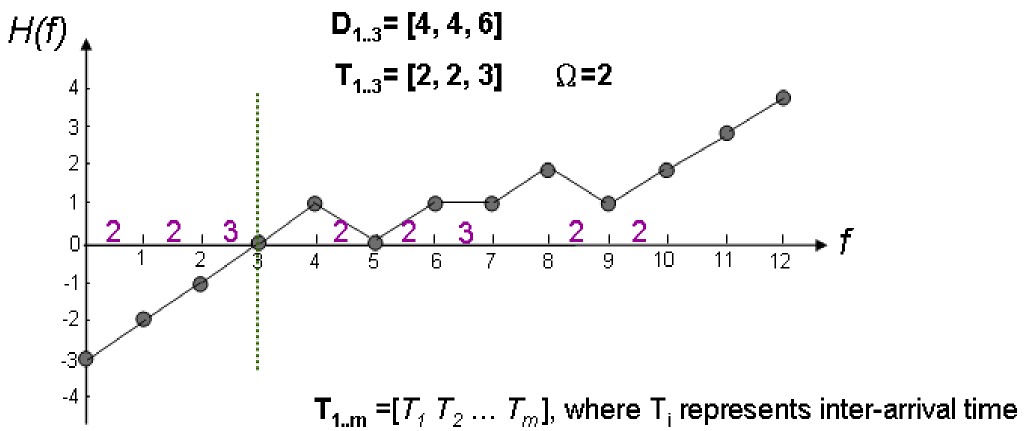


Figure 5.3 An example of the empty-minus-backlog curve

It is shown in [2] that $H(f)$ is equal to the number of empty frames up to frame f minus the backlog at the end of frame f. That is to say, if $E(f)$ and $B(f)$ denote, respectively, the number of empty frames up to frame f and the backlog at the end of frame f, then we have $H(f) = E(f) - B(f)$.

Moreover, connection m is schedulable iff the curve $H(f)$ ever hits the x-axis up to frame D_m, where D_m represents the delay bound of the connection m. (i.e., connection m has to be served before the end of frame D_m.) Similarly, connection k is schedulable iff the curve $H(f)$ ever hits $-m+k$ up to frame D_m. Notice that a connection k is schedulable implies its packets would be served no later than frame D_k. As in figure 7, for connection 3, it has to be served before the end frame 3, which implies the curve $H(f)$ has to hit the x-axis before frame 3. On the other hand, for the other two connections, we can verify the descriptions as well. This can be shown as follows, $\mathbf{D_{1..m}}$ is schedulable iff there exists an f_k, $1 \le f_k \le T_k$, such that $H(f_k) \ge -m+k$ for all k, $1 \le k \le m$.

5.3 The minimum inter-arrival time of next acceptable connection

Also, from the curve $H(f)$, we can derive the minimum inter-arrival time of a connection that can be added to the existing *m* connections. There are two topics that we should concern about here. One topic is that the new connection should not to violate the schedulability of these existing *m* connections. The other is the new connection must be schedulable in the system, which is already proven in [2].

We consider the effect of adding a new connection with inter-arrival time *T* to connection *m* first. Suppose there is more than one empty frame before the end of frame D_m. And we say frame *f* is a maximum (of connection *m*) iff it is an empty frame and $H(f) \geq H(f+1)$. It is shown in [2] that if one can move the packets to an empty frame, in order to maintain the schedualbility of connection *m*, the minimum period of the new connection is equal to $\lceil f / H(f) \rceil$. Moreover, it is proven that we only need to consider the maxima before D_m. Therefore, in our case that $D = \Omega T$, if we consider one of the maxima, the minimum inter-arrival time will be $\lceil \lceil f / H(f) \rceil / \Omega \rceil$. We can get the following statement.

If there are *i* maxima $f_1, f_2, \ldots$, and f_i up to frame D_m, then

$$T_{next,m} = \min_{1 \leq k \leq i} \left\{ \left\lceil \frac{\lceil f_k / H(f_k) \rceil}{\Omega} \right\rceil \right\} \tag{4.2}$$

Figure 8 shows another example of $H(f)$. In this figure, connection 5 (suppose the connection number increases from the left to the right) has to be served before the end of frame 12. There are two maxima in this case. Hence, in order to maintain the schedulability of connection 5, $T_{next,5} = 1$.

In the case that there is no empty frame before the end of $D_{m.}$ Suppose that the first empty frame occurs at f_e. The minimum inter-arrival time of next acceptable connection will be $T_{next,m} = \lceil f_e / \Omega \rceil$. Figure 9 shows an example of this kind of case. The first empty frame occurs at the end of frame 4. Hence, $T_{next,3} = 2$.

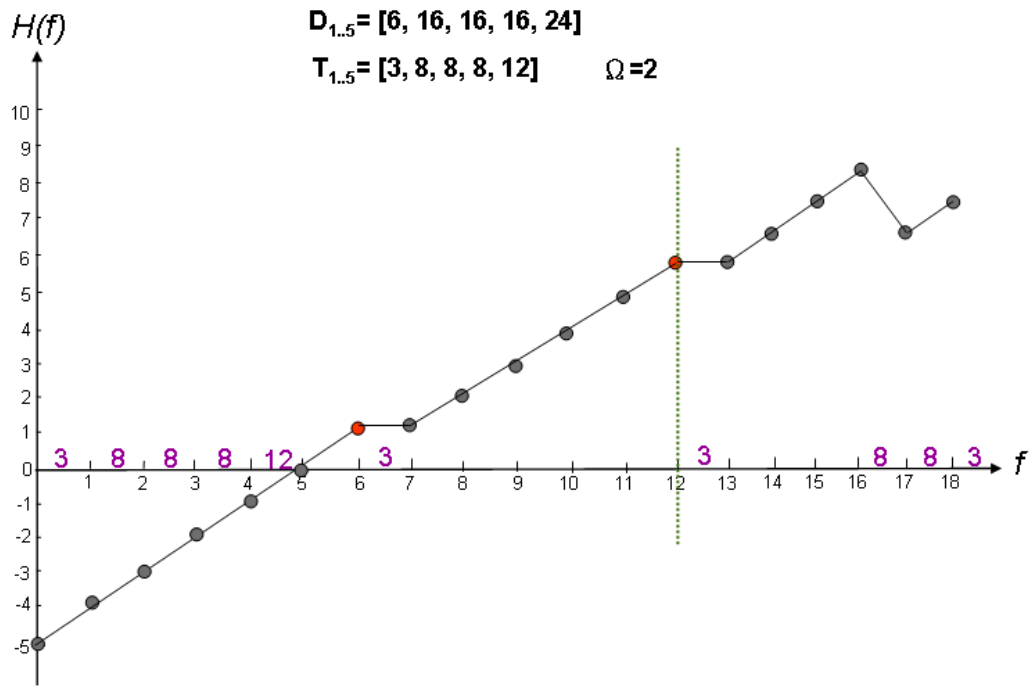


Figure 5.4 $H(f)$ **for** $D_{1..5}$ **= [6, 16, 16, 16, 24]**

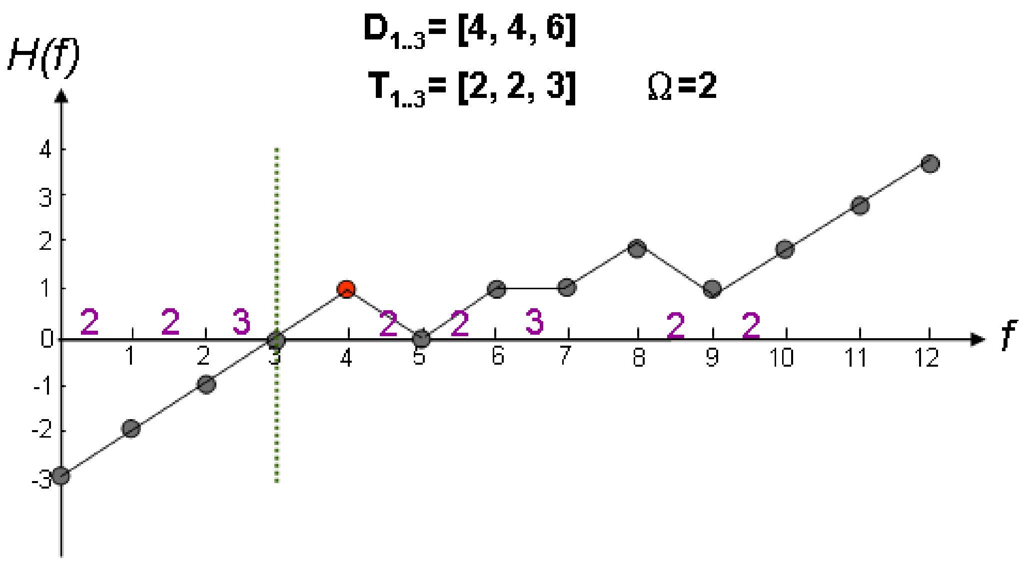


Figure 5.5 ***H*(*f*) for D$_{1..3}$ = [4, 4, 6]**

Let us consider the effect of adding a new connection to connection $m-1$. Similarly, we can find $T_{next,m-1}$, the minimum inter-arrival time of the new connection subject to the constraint that connection $m-1$ is served no later than frame T_{m-1}. A maximum now is defined as a frame f which satisfies $1 \le f \le T_{m-1}$, $H(f) > 0$, $H(f) > H(g)$ for all $g < f$ and $H(f) \ge H(f+1)$. We can obtain the following formula:

$$T_{next,m-1} = \min_{1\le k\le i}\left\{ \left\lceil \frac{\left\lceil \frac{f_k}{H(f_k)+1} \right\rceil}{\Omega} \right\rceil \right\} \qquad (4.3)$$

Notice that it has $H(f)+1$ in the formula, which is different from the case when we consider connection *m*. This is because here in the case $m-1$, we only discuss about the schedulability about itself.

Also, if there no maximum up to T_{m-1}, then set $T_{next,m-1} = \lceil f_0 / \Omega \rceil$, where f_0 is the first frame where $H(f) = 0$.

Finally, we can find $T_{next,j}$, for all *i*, $1 \le i \le m$. A maximum is modified as a frame which satisfies $1 \le f \le T_i$, $H(f) > -m + i + 1$, $H(f) > H(g)$ for all $g < f$ and $H(f) \ge H(f+1)$. Then, we can get $T_{next,j}$ as follows,

$$T_{next,j} = \min_{1 \le k \le i} \left\{ \left\lceil \frac{\left\lceil \frac{f_k}{H(f_k) + m - j} \right\rceil}{\Omega} \right\rceil \right\} \tag{4.4}$$

Now, we have known all the inter-arrival time of next acceptable connection without violating the scheduability of each connection. The value T_{next} will be like below,

$$T_{next} = \max_{1 \le j \le m} \left\{ T_{next,j} \right\} \tag{4.5}$$

Therefore, Theorem 4 in [2] can be re-written as follows,

If $H(f) \le 0$, for all *f*, $1 \le f \le C$, then no more connection can be accepted in the system. If $H(f) > 0$, for some $f \le C$, then the minimum inter-arrival time of next acceptable connection is given by $T_{next} = \max_{1 \le j \le m} \left\{ T_{next,j} \right\}$

Chapter 6

Numerical Results

In this section, we show our simulation results. There are four evaluation metrics in our simulation, which are the average sleeping cycle, channel occupancy, bandwidth efficiency and the number of serviced MSSs.

The average sleeping cycle is measured as the sleeping interval plus listening interval. The channel occupancy implies the ratio of the bandwidth used in the whole simulation. The bandwidth efficiency shows the average ratio of the bandwidth used in a frame.

6.1 Environment setup

The second type of power saving class defined in IEEE 802.16e is adopted here. We only consider the downlink scheduling for UGS traffic. Each MSS has only one connection. Refer to the ITU Telecommunication Standardization Sector (ITU-T) for VoIP, the followings are the parameters what we use in this simulation.

We divide our simulation into two parts. The first part is evaluated with one connection of type 1 and *m* connections of type 2. We compare our proposed methods to MMPS and MMPS-BF. Proposed A is each connection becomes eligible if it accumulates Ω in the queue, while proposed B is $\Omega - 1$. The second part is evaluated with *m* connections of type2 , type 3, and type 4. We compare these methods as well.

Inter-arrival rate (type 1)	15ms
Inter-arrival rate (type 2)	30ms
Inter-arrival rate (type 3)	60ms
Inter-arrival rate (type 4)	90ms
Delay constraint (type 1)	150ms
Delay constraint (type 2)	300ms
Delay constraint (type 3)	600ms
Delay constraint (type 4)	900ms
OFDM frame length	5ms
Frame capacity (Ω)	10

Table 6.1 Simulation parameters

6.2 Numerical results

Figure 6.1, 6.2, 6.3 are the simulation of the first part, and figure 6.4 and 6.5 are of the second part. In figure 6.1, the amount of sleeping cycle of MMPS and MMPS-BF are all the same. This is because they are bounded in a constant sleeping cycle, which is called the common sleeping cycle of all connections. The amount of proposed A (Ω case) and proposed B increases while the number of type 2 MSS increases. However, proposed A only can accommodate 5 connections of type 2, the curve would not rise up after 5 connections. This is because that all the connections are supposed to be served before the head of line in the queue expires. Similarly, one can observe that proposed B would not rise up after 11 connections due to there is no more capacity for any other connection to add in.

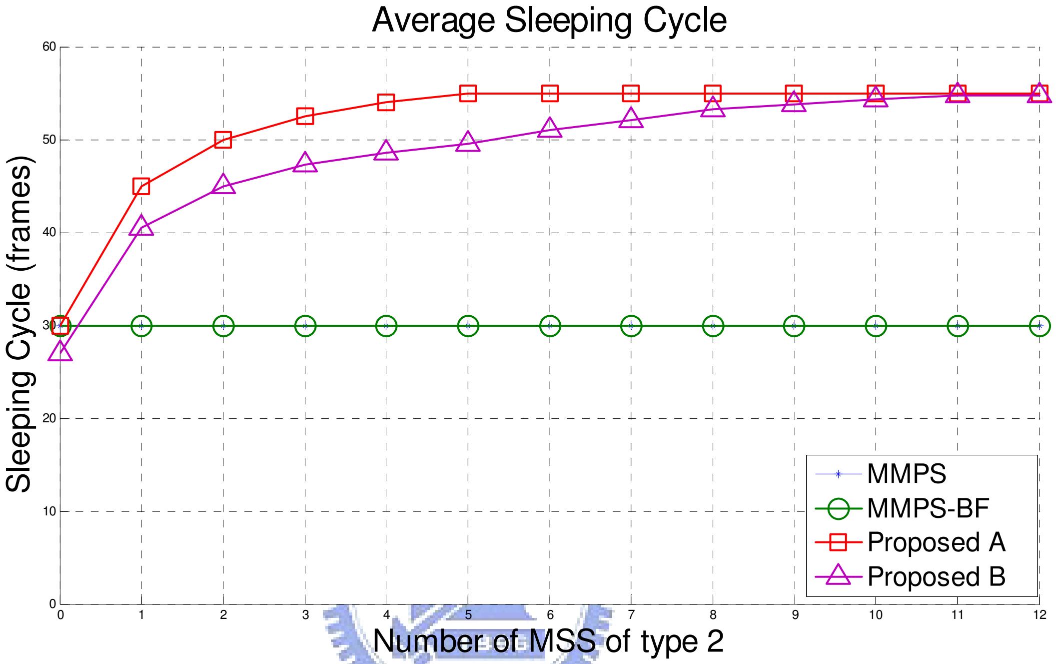


Figure 6.1 The average sleeping cycle

Figure 6.2 shows the channel occupancy. We can observe that all the curves are tend to rise up while the number of type 2 MSS increases. The ratio of MMPS is getting higher with the increase of number of stations. Almost every other adjacent two points are at the same value in the curve of MMPS-BF. The reason is that each type 2 MSS occupy half of a frame now. Besides, the two proposed methods are going up from the beginning. Due to the same reason described before, proposed A maintains the same ration after 5 connections, and so does proposed B after 11 connections.

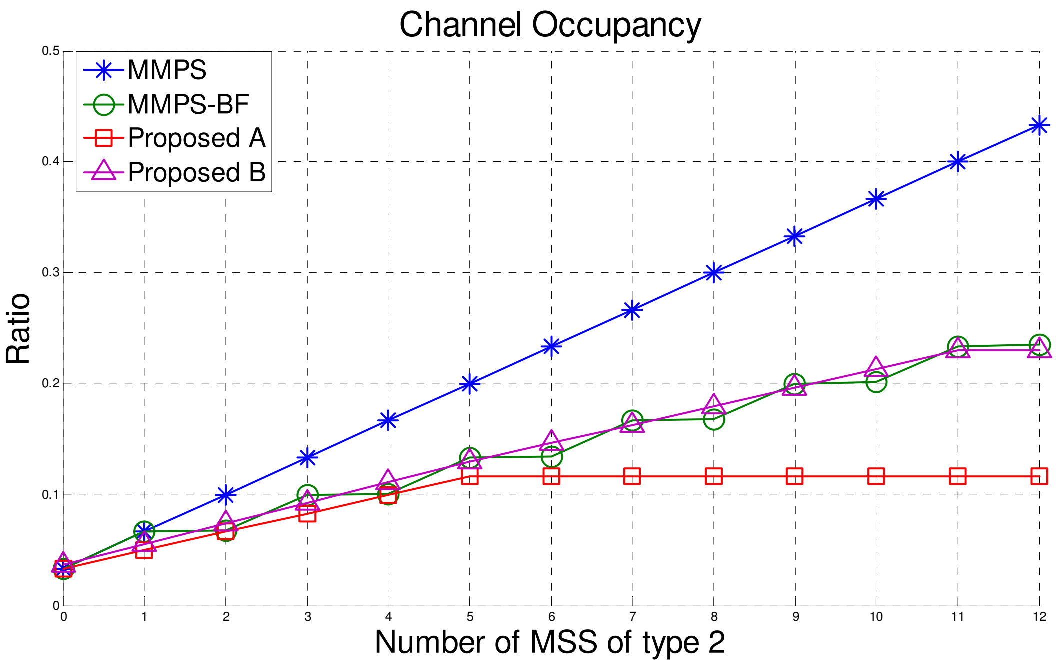


Figure 6.2 Channel Occupancy

In figure 6.3, it shows the bandwidth efficiency. Proposed A obtains the best bandwidth efficiency with the ratio is equal to 1. This is because the each connection would only be served while its queue is full of Ω packets, and Ω is also the frame capacity. Proposed B performs worse than A by less than 0.1. The reason is that Ω-1 would not fully utilize the bandwidth of a frame. The curve of MMPS goes down while the number of type 2 MSS increases. The reason is that when there is no type 2 connection, the common sleeping cycle would be the sleeping cycle of type 1, and it would definitely be served Ω packets per time. But with adding type 2 MSS in the system, the average bandwidth efficiency would get lower due to type 2 can not fully utilize its bandwidth. As what we explain before, in MMPS-BF, each connection occupies half of a frame so that the points would be equal to 1 every other two adjacent connections.

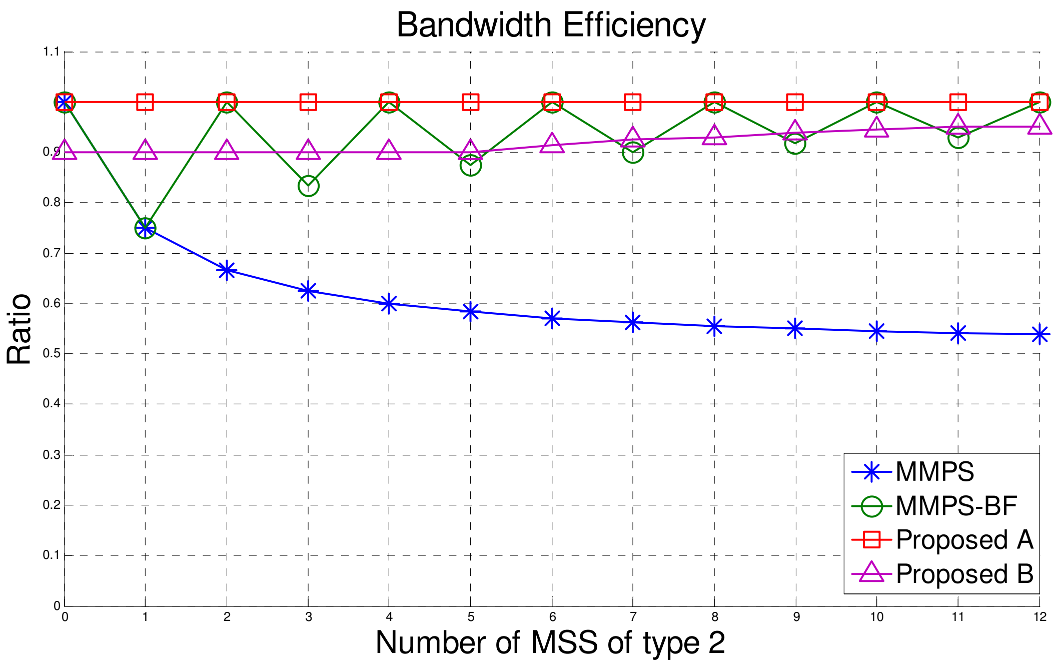


Figure 6.3 The bandwidth efficiency

Figure 6.4 and figure 6.5 show the second part of our simulation. In figure 6.4, because MMPS and MMPS-BF have a common sleeping cycle, the number of serviced MSSs would be bounded at 30 here. However, Proposed A only can accommodate 3 connections ($m = 1$, i.e. one type 2, one type 3, and one type 4 connections.) while proposed B accommodate 6 connections. ($m = 2$, i.e. two type 2, two type 3, and two type 4 connections.)

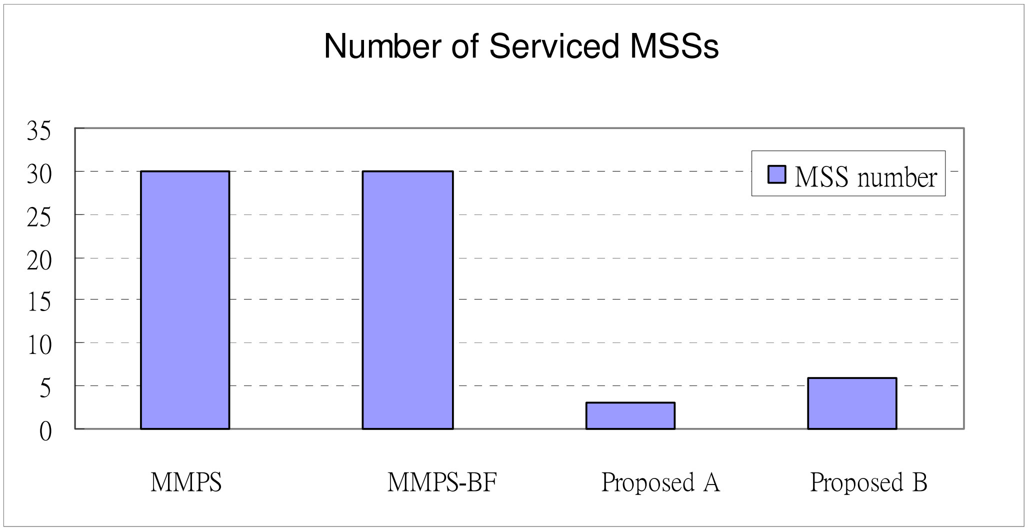


Figure 6.4 Number of serviced MSSs

Figure 6.5 shows another point of view of bandwidth efficiency. Suppose that the simulation here would not violate the system capacity. Proposed A achieves 1 of the bandwidth efficiency. Proposed B achieves 0.9 due to $\Omega-1$ is equal to 9. The ratio of MMPS-BF is almost the same as proposed B. Besides, MMPS performs the worst among the four methods.

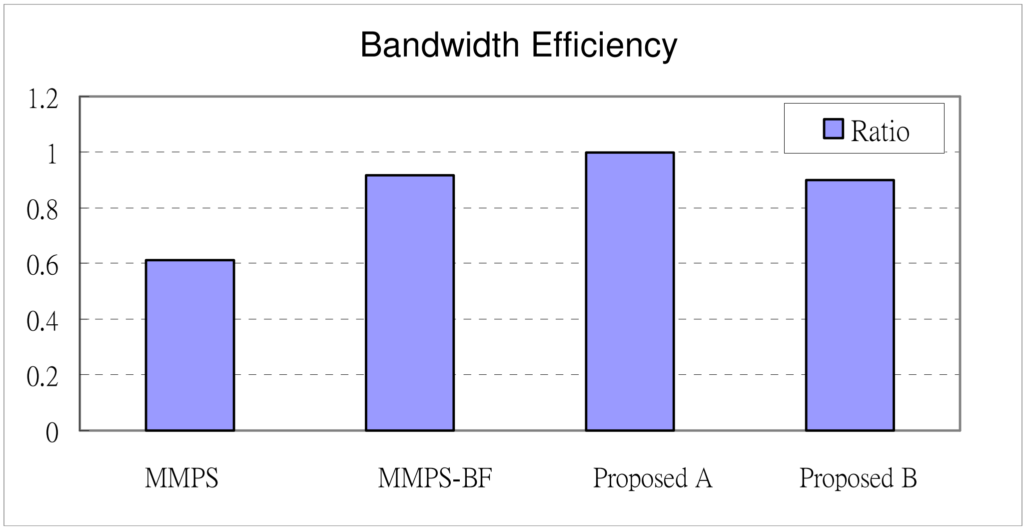


Figure 6.5 Bandwidth efficiency

From the results above, we can conclude that Proposed A achieves longer sleeping cycle and better bandwidth efficiency than the other three methods, but it performs the worst throughput. Obviously, there is a tradeoff between bandwidth efficiency and system throughput.

Chapter 7

Conclusions

This thesis proposes a scheduling algorithm with a system model under the assumption that $D = \Omega T$. We derive the admission control of this algorithm as well. The main idea of the admission control is to find the minimum inter-arrival time of next acceptable connection.

Our proposed method performs well in the average sleeping cycle and the bandwidth efficiency, but achieves worse system throughput than MMPS and MMPS-BF. In this paper, we only discuss the theoretical system model of Ω case. In the future, we will introduce $\Omega - k, k = 1 \sim \Omega - 1$ into our model. Furthermore, we will discuss the case of $D \neq \Omega T$.

Bibliographies

---

[1] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in hard real-time environment," Journal of ACM, vol.20, pp.44-61, Jan. 1973.

[2] T. H. Lee and A. B. Chang, "Fast admission control for rate monotonic schedulers," IEICE Trans. Commun., Vol.E82-B, No.1 Jan. 1999.

[3] S. C. Huang, R. H. Jan, and C. Chen, "Energy efficient scheduling with QoS guarantee for IEEE 802.16e broadband wireless access networks," In proceedings of IEEE International Wireless and Mobile Computing Conference 2008.

[4] IEEE Std 802.16e, "IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems," Feb. 2006.

[5] J. G. Andrews, A. Ghosh, and R. Mohamed, "Fundamentals of WiMAX, Understanding Broadband Wireless Networking," Prentice Hall Feb. 2007.

[6] Y. L. Chen and S. L. Tsao, "Energy-efficient Sleep Mode Operations for Broadband Wireless Access Systems," In proceedings of IEEE Vehicular Technology Conference 2006.

[7] Dinh Thi Thuy Nga, and M.G. Kim, "Delay-guaranteed Energy Saving Algorithm for the Delay-sensitive Applications in IEEE 802.16e systems," IEEE Transactions on

Consumer Electronics, 2007.

[8] A. K. Parekh, and R. G. Gallager, “A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case,” IEEE Transactions on Networking, 1993.

[9] Vandana Singh and Vinod Sharma, “Efficient and Fair Scheduling of Uplink and Downlink in IEEE 802.16e OFDMA Networks,” In proceedings of IEEE Wireless Communication and Networking, 2006.

[10] Q. Zhao and Danny H. K. Tsang, “An Equal-Spacing-Based for QoS Guarantee in IEEE 802.11e HCCA Wireless Networks,” IEEE Transactions on Mobile Computing Nov., 2008.

[11] J Jang, K Han, S Choi, “Adaptive Power Saving Strategies for IEEE 802.16e Mobile Broadband Wireless Access,” IEEE Asia-Pacific Conference on Communications, 2006.

[12] J Shi, G Fang, Y Sun, J Zhou, Z Li, and E Dutkiewicz,”Improving mobile station energy efficiency in IEEE 802.16e WMAN by burst scheduling,” IEEE Globecom, 2006.

[13] L Kong, Danny H.K. Tsang, “Optimal Selection of Power saving classes in IEEE 802.16e,” In Proceedings of IEEE Wireless Communications and Networking Conference, 2007.

[14] L. Tian, Y. Yang, J. Shi, E. Dutkiewicz, and G Fang, ”Energy Efficient Integrated Scheduling of Unicast and Multicast Traffic in 802.16e WMANs,” IEEE Globecom, 2007.

[15] S. Zhu, T. Wang, “Enhanced Power Efficient Sleep mode operation for IEEE 802.16e based WiMAX,” IEEE Mobile WiMAX Symposium, 2007

[16] Y. Zhang, “Performance Modeling of Energy Management Mechanism in IEEE 802.16e

Mobile WiMAX," In Proceedings of IEEE Wireless Communications and Networking Conference, 2007.

[17] Y Ge, GS Kuo, "An efficient sleep mode management scheme in IEEE 802.16e networks," IEEE International Conference on Communications, 2007.

[18] X. Perez-Costa, D Camps-Mur, "AU-APSD: Adaptive IEEE 802.11e Unscheduled Automatic Power Save Delivery," IEEE International Conference on Communications, 2006.