國立交通大學

資訊學院資訊科技(IT)產業研發碩士班

碩士論文

在 IEEE802.16 WiMAX 網路中要求頻寬機制

對 Best Effort 連線的影響

The Impact of Bandwidth Request Mechanism on Best

Effort Connections in IEEE 802.16 WiMAX Networks

研 究 生：張毓庭

指導教授：趙禧綠　助理教授

中華民國九十八年七月

在 IEEE802.16 WiMAX 網路中要求頻寬機制

對 Best Effort 連線的影響

The Impact of Bandwidth Request Mechanism on Best Effort Connections in IEEE 802.16 WiMAX Networks

研 究 生：張毓庭　　Student：Yu-Ting Chang

指導教授：趙禧綠　　Advisor：His-Lu Chao

國 立 交 通 大 學

資訊學院資訊科技(IT)產業研發碩士班

碩 士 論 文

A Thesis
Submitted to Institute of Network Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Industrial Technology R & D Master Program on
Computer Science and Engineering

July 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年七月

摘要

異質網路的排程是一個越來越流行的議題，我們可以發現有越來越多的相關文獻被提出，而在這些文獻中，作者們大多都是使用選擇傳送時間最短的路徑來做排程的動作。但是這些作者們都忽略了不同的要求頻寬機制(bandwidth request mechanism)會造成的影響。要求頻寬機制會影響上行鏈路(uplink)的頻寬分配(bandwidth allocation)，然而這些作者們僅考慮了不同網路介面的資料速率(data rate)，而沒考量要求頻寬機制的影響，如此一來，使用他們的方法並無法正確估算封包在 IEEE 802.16 WiMAX 網路中的傳輸時間。因此，我們研究了在 IEEE 802.16 WiMAX 網路中要求頻寬機制對 Best Effort 連線造成的影響。在此篇論文中，我們會討論在 Best Effort 連線上不同的要求頻寬機制，且有額外討論兩種都是以競爭(contention)為基礎的要求頻寬機制及兩種都是以揹負式(piggyback)為基礎的要求頻寬機制。從模擬的結果中，我們可以得知在網路流量大時，可使用輪詢(polling)的機制來要求頻寬；而在網路流量小時，可使用競爭(contention)機制或揹負式(piggyback)機制來要求頻寬。我們的研究結論可以幫助 IEEE 802.16 WiMAX 媒體存取控制(MAC)層排程機制的設計者設計出更有效率獲得頻寬的排程方式來增進系統的效能。

Abstract

Heterogeneous network scheduling becomes more and more popular. There are some papers do the research of heterogeneous network scheduling, and they use the concept that choosing the interface with the shortest transmission time. However, in those papers, they didn't consider the impact of bandwidth request mechanisms. In uplink, bandwidth request mechanisms affect the bandwidth allocation, and in those papers, they cannot use their methods to estimate the delivery time in IEEE 802.16 WiMAX networks, because they just consider the data rates of different interfaces, they didn't consider the effects of the bandwidth request mechanisms. So we study the impacts of bandwidth request mechanisms for Best Effort (BE) traffic in IEEE 802.16 WiMAX networks. In this thesis, we discuss the different bandwidth request mechanisms on Best Effort connections, and discuss two different contention-based mechanisms and two different piggyback-based mechanisms additionally. From the simulation results, we can know the conclusion that in heavy traffic load, we should use polling mechanism; in light traffic load, we should use contention mechanism and piggyback mechanism. By our study, the conclusion may help the IEEE 802.16 WiMAX MAC layer scheduler designers to design a more efficient scheduler for requesting bandwidth to increase the system performance.

致謝

時光匆匆流逝，兩年就這樣過去了，求學生涯也到此暫時告一個段落。回想這兩年的點點滴滴，確實在我的人生歷練中，無論是在學業方面或是人際關係上又多刻劃了相當多筆的痕跡。

在還未當上研究生之前，我對做研究的難度並沒有任何的體認。直到真的開始做研究後，才發現做研究並不是一件很簡單的事情。

在這段過程中，我遭遇到了不少的困難，所幸在我每每感到挫敗的時候，都有人會適時的給予我幫助與鼓勵，並在我最低潮的時候陪伴我，讓我不至於被研究給打倒，在此謝謝所有曾經幫助過我的人。

這兩年來，除了學習如何做研究之外，最大的收穫是遇到了瞭解彼此的同伴，除了指導教授的督導外，若是沒有你的幫忙，這篇論文是不可能這麼快誕生的，感謝你不時的提出建議，更感謝你在百忙之中，還可抽空聽我訴說心事，謝謝你。

最後，能完成這本論文，首要感謝指導教授在這段時間內的鞭策，也感謝您並沒有在我逃避的時候放棄我，而是適時的點醒我，讓我得以完成此篇論文；感謝實驗室學姐與同學們的關心與幫助，跟你們一起討論真的受益良多，而你們付出的關心，更讓我感受到實驗室的溫暖，謝謝你們陪我度過這兩年的光陰；也感謝家人與朋友們的支持與鼓勵，沒有你們，就沒有現在這篇論文。

讓我對你們致上最誠懇的感謝之心，謝謝大家！

Contents

List of Figures

List of Tables

Chapter 1. Introduction

Heterogeneous network scheduling becomes more and more popular. Several papers concern about this issue. In [6, 7], the authors propose an algorithm named EDPF (Earliest Delivery Path First) to do the heterogeneous network scheduling. They calculate the packet delivery time for each path, and select the path with shortest packet delivery time to transmit data. There are some other papers also do the research of heterogeneous network scheduling, and their main ideas are also from [6, 7]. In [8, 9], the authors modify the EDPF method in [6] using time-slotted concept, and in [10], the authors add the concept that the packet transmission may be fail into EDPF method in [6]. In those papers, they do not think about the effect of bandwidth request mechanisms. In uplink, bandwidth request mechanisms affect the bandwidth allocation, and in those papers, they cannot use their methods to estimate the delivery time in IEEE 802.16 WiMAX networks, because they just consider the data rates of different interfaces, they do not consider the effects of bandwidth request mechanisms. Though we can add the bandwidth request mechanism concept in the above work, we find another problem. In WiMAX networks, if the Best Effort (BE) connection using the contention-based mechanism to get grant from the BS, the connection should contend for the next generated packet (which is not be scheduled) after all the scheduled packets are transmitted, just like Fig. 1. In Fig.1, the SS requests bandwidth for packet A in Frame I, and packet A transmits in Frame (I+1). Packet B is arrived in Frame (I+1), but the SS cannot request bandwidth for packet B, it should wait until Frame (I+2) to send the request for packet B. This means that if the SS can get grant every time, the SS transmits the bandwidth request once at least two frames (the first frame is for sending the bandwidth request, and the second frame is for sending packets). This causes the long bandwidth request delays, and this may let the scheduling in heterogeneous networks become unfair.

There is another reason for us to study the effects of bandwidth request mechanisms for BE traffic in IEEE 802.16 WiMAX networks. There are some papers discussing about the analysis of contention periods in IEEE 802.16 WiMAX networks. In these papers, the authors propose equations to determine the length of contention periods. In [11], the authors finally say that the best contention period size is 2*N*-1 (*N* means the number of SSs). Though they determine the best length of contention periods, in heavy traffic loads, the BS still cannot grant the SS because of the inefficient bandwidth (the longer contention period, the shorter uplink subframe). So it is no use to larger the contention period size when the bandwidth is inefficient, and this is why we want to know the effects of bandwidth request mechanisms for BE traffic in IEEE 802.16 WiMAX networks.

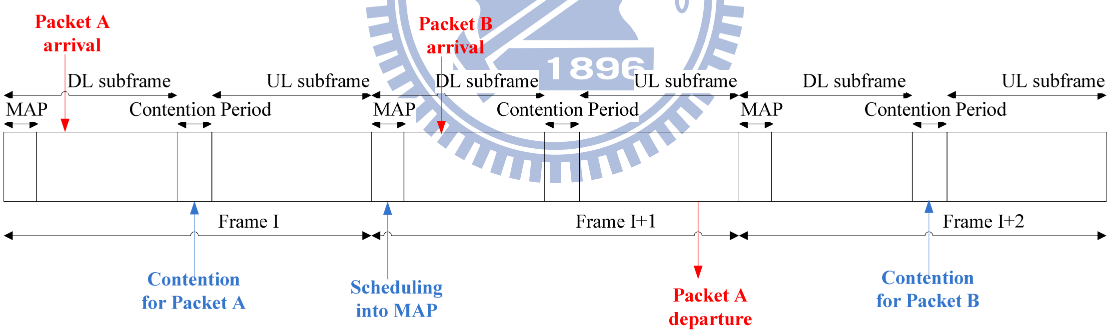

Fig. 1 Contention mechanism in IEEE 802.16 WiMAX network

1.1 An Overview of IEEE 802.16 WiMAX Network

The IEEE 802.16 WiMAX network is for metropolitan broadband wireless access (BWA) systems, and there is an industrial association called Worldwide Interoperability for Microwave Access Forum was formed to improve the standards by specifying specifications between IEEE 802.16 network products from different sellers. Accordingly, IEEE 802.16 networks are referred to WiMAX networks.

In the IEEE 802.16 standard [1], it said that WiMAX interfaces can support two kinds of topologies, one is PMP (Point-to-Multipoint) topology and the other is mesh topology. In the PMP network, a centralized base station (BS) controls the resources between the subscriber stations (SSs) and the BS. While in the mesh network, network traffics can be routed through other SSs or directly transmitting between SSs. In this thesis, we only concern about the PMP mode, and it is the first choice of WiMAX operators.

The MAC layer in IEEE 802.16 WiMAX networks is connection-oriented. Shortly after the SSs do the registration, connections are combined with different service flows (one connection per service flow), and different service flows define the different QoS parameters for the MAC PDUs. Service flows provide the mechanisms for uplink and downlink QoS management, and they are integral to the bandwidth allocation process in IEEE 802.16 WiMAX network MAC layer.

IEEE 802.16 WiMAX network MAC layer uses a request/grant mechanism to control transmissions. If a SS needs to transmit data to the BS, it should first request an opportunity to transmit data from the BS, and then the BS is responsible to allocate a transmission opportunity called grant in the next uplink subframe. One SS may request uplink bandwidth on a per connection basis, but the BS grants the bandwidth to SSs on a per SS basis.

There are different types of uplink bandwidth requests, polling, contention, bandwidth stealing, and piggyback requests. Except the unsolicited grants mechanism is provided the bandwidth request by the BS actively, the other mechanisms need SSs to send bandwidth requests for getting grant from the BS. We will give more detailed descriptions for these types of uplink bandwidth request in the section 1.1.1.

Scheduling services represent mechanisms to handle data supported by the IEEE 802.16 WiMAX network MAC layer scheduler for data transmissions on a connection (service flow). Each connection is related to one data service, and each data service is related to a set of QoS parameters. There are four services are supported: Unsolicited Grant Service (UGS), Real-time Polling Service (rtPS), Non-real-time Polling Service (nrtPS), and Best Effort (BE). section 1.1.2 gives more detailed description about these four services.

1.1.1 Bandwidth Request Mechanisms

In IEEE 802.16 standard [1], there are different types of uplink bandwidth requests, polling, contention, bandwidth stealing, and piggyback requests. Except the unsolicited grants mechanism is provided the bandwidth request by the BS actively, the other mechanisms need SSs to send bandwidth requests for getting grant from the BS. In this section, we will introduce the different bandwidth request mechanisms, polling, contention, bandwidth stealing, and piggyback.

1.1.1.1 Polling Mechanism

Polling is the process that the BS allocates bandwidth to the SSs for the purpose that SSs can send bandwidth requests to the BS, and polling is also done on the SS basis. There are three types of polling, unicast polling, multicast polling, and broadcast polling.

First, let us discuss about the unicast polling. Once a SS is polled individually, the SS will be allocated in the UL-MAP, and the allocated bandwidth is sufficient for the SS to send one bandwidth request. If a SS has an UGS connection, the SS should not be polled by the BS unless the PM bit is set in the Grant Management subheader.

Then, we talk about the multicast polling and broadcast polling. The difference among multicast polling, broadcast polling, and unicast polling is that the BS allocates bandwidth to the different connection identifiers (CIDs). In multicast polling and broadcast polling, the BS allocates bandwidth to the multicast CIDs and broadcast CIDs, while in the unicast polling, the BS allocates bandwidth to the SS's basic CID. In addition, multicast polling and broadcast polling are used when there is insufficient bandwidth to individually poll all SSs. In this situation, some SSs may be polled in multicast groups or the broadcast poll may be issued. In this thesis, we just discuss about the unicast polling. There is a simple example for unicast polling in section 2.2.

1.1.1.2 Contention Mechanism

Contention may occur in the request intervals (contention periods), once the SSs want to send bandwidth requests, the SSs will do the contention resolution in the contention period at the current frame.

The mandatory method of contention resolution should be supported is the truncated binary exponential backoff with the initial backoff window and the maximum backoff window controlled by the BS. The values represent a power-of-two value, that is, a value of 4 means a window between 0 and 15, and a value of 6 means a window between 0 and 63.

The SSs should select a random number in its backoff window, and this random number means the number of contention transmission opportunities that the SSs should defer before transmitting its bandwidth request. After a bandwidth request transmission, if the SS can get grant at the next frame, the contention resolution is complete, and if the SS cannot get grant at the next frame, the SS should double its backoff window, as long as the value is less than the maximum backoff window, and continuing the contention resolution process. In addition, there is a simple example for contention mechanism in section 2.1.

1.1.1.3 Bandwidth Stealing Mechanism

Bandwidth stealing mechanism is also one of the methods for the SSs to send bandwidth requests. In bandwidth stealing mechanism, the SS will use a portion of bandwidth that had allocated for the SS to transmit data to send a bandwidth request.

1.1.1.4 Piggyback Mechanism

Piggyback mechanism is to use the MAC subheader to request additional bandwidth for the current connection, and we call this request as the piggyback request. Once the SS has one packet can be transmitted in this frame, the SS can request bandwidth for this connection by piggybacking a bandwidth request in the scheduled packet MAC subheader. There is a simple example for piggyback mechanism in section 2.3.

1.1.2 Scheduling Services

The following are the brief introduction of the four scheduling services, Unsolicited Grant Service (UGS), Real-time Polling Service (rtPS), Non-real-time Polling Service (nrtPS), and Best Effort (BE).

1.1.2.1 Unsolicited Grant Service (UGS)

The UGS is designed for real-time service flows that generate fixed size packets on a periodic interval. The BS should provide grants to the SSs at periodic intervals based on the Maximum Sustained Traffic Rate of the service flow, and the SSs are prohibited from using any contention requests for these connections. There is a Grant Management subheader used to pass the status information from the SS to the BS regarding the state of the UGS flow, and in this subheader, there is a field called poll-me (PM) bit is used to request to be polled for a different, non-UGS connection. On the other words, UGS is allowed to request a unicast poll for bandwidth need by non-UGS connections if the PM bit is set. In addition, UGS is not allowed to request bandwidth using piggyback requests and bandwidth stealing.

The QoS parameters hold by UGS are the Maximum Sustained Traffic Rate, the Minimum Reserved Traffic Rate (if this presents, it should have the same value as the Maximum Sustained Traffic Rate), the Maximum Latency, the Tolerated Jitter, and the Request/Transmission Policy.

1.1.2.2 Real-time Polling Service (rtPS)

The rtPS is designed for real-time service flows that generate variable size packets on a periodic interval. The BS should provide periodic unicast request opportunities for the SSs to request bandwidth, and the SSs are prohibited from using any contention requests for these connections. The rtPS is allowed to request bandwidth by piggyback requests, bandwidth stealing, and unicast polling.

The QoS parameters hold by rtPS are the Maximum Sustained Traffic Rate, the Minimum Reserved Traffic Rate, the Maximum Latency, and the Request/Transmission Policy.

1.1.2.3 Non-real-time Polling Service (nrtPS)

The nrtPS is designed for delay-tolerant service flows that generate variable size packets and a minimum data rate is required. The BS provides timely unicast request opportunities for the SSs to request bandwidth, and the SSs are allowed to use contention requests for these connections. The nrtPS is allowed to request bandwidth by piggyback requests, bandwidth stealing, and polling.

The QoS parameters hold by rtPS are the Maximum Sustained Traffic Rate, the Minimum Reserved Traffic Rate, the Traffic Priority, and the Request/Transmission Policy.

1.1.2.4 Best Effort (BE)

The BE is designed for the service flows which no minimum service level is required, and it is handled on a space-available basis. The BE is allowed to request bandwidth by piggyback requests, bandwidth stealing, and polling.

The QoS parameters hold by rtPS are the Maximum Sustained Traffic Rate, the Traffic Priority, and the Request/Transmission Policy.

Now we have introducing how does the IEEE 802.16 WiMAX network MAC layer scheduler operate, and we also have introducing the bandwidth request mechanisms and the different scheduling services. Next we will show the related work about this thesis.

1.2 Related Work

In [3], the authors propose an approach comparing contention based and polling based bandwidth request methods, and they also provide an analytical model for the contention based bandwidth request method. They focus on the bandwidth request delays between the different methods, but they do not consider the packet transmission delays. In our work, we consider both the bandwidth request delays and packet transmission delays. In [4], the authors do the research about the piggyback request mechanism. They show that using the piggyback request mechanism can reduce the collisions of bandwidth requests, and the performance increase by using piggyback requests when there are a lot of users and all of them are with short packet inter-arrival times. In [5], the authors study the contention based and polling bandwidth request mechanisms. The major difference with [3] is the error-prone channel, that is, they simulate contention based bandwidth request mechanism with both error-free and error-prone channels. It is the same as [3] that they only consider the bandwidth request delays and do not consider the packet transmission delays, too. In [11], [12], [13], all of them show the optimal size of contention periods for the contention-based bandwidth request mechanism, they use mathematical model to find the optimal contention period size, but once the contention period becomes longer, the uplink capacity will be shorter. So we do other bandwidth request mechanisms in this thesis for saving the time of bandwidth request delays.

We concentrate on contention/polling/piggyback bandwidth request mechanisms at the same time, and there is no paper has been published studying these mechanisms at once. In this thesis, we show the impacts of the contention/polling/piggyback bandwidth request mechanisms for BE traffic in IEEE 802.16 WiMAX networks.

1.3 Organization

The organization of the following thesis is as follows. Chapter 2 describes the bandwidth request mechanisms on Best Effort connections in IEEE 802.16 WiMAX network, Chapter 3 describes the proposed bandwidth request mechanisms on Best Effort connections in IEEE 802.16 WiMAX network, Chapter 4 describes our simulation environment and results, and Chapter 5 is the conclusions and future work of this thesis.

Chapter 2. Bandwidth Request Mechanisms for Best Effort Connections

In the IEEE 802.16 standard [1], BE connections can use contention mechanism, polling mechanism, piggyback mechanism, and bandwidth stealing mechanism to request bandwidth. We have already introduced the above bandwidth request mechanisms in Chapter 1. In fact, the standard do not illustrate how to realize these mechanisms, here, we discuss about the following mechanisms, contention-based bandwidth requests, piggyback requests, or polling to get bandwidth for BE connections from the BS.

In this chapter, we will show how to use contention, polling, and piggyback mechanisms sending bandwidth requests and give figures to illustrate.

2.1 Type I - Contention Mechanism

In Type I - Contention mechanism, if there are BE packets need to be sent by the SSs, the SSs will send a bandwidth request to the BS in the contention periods. First, the SSs will randomly choose a number between 0 to W_0-1 (W_0 is the initial contention window size). Then, the SSs will send bandwidth requests to the BS after the random choosing slot time in the contention periods. Once the SSs are failed to transmit a bandwidth request to the BS, the SS will double the contention window size like the IEEE 802.11 backoff procedure dose (truncated binary exponential backoff procedure). If the transmission succeeds, the contention window size should be reset to W_0.

In Fig. 2(a), there is one BE packet arrives in Frame I, and the SS does the contention resolution as mentioned above to send a bandwidth request in Frame I's contention period. In Frame (I+1), the SS knows that it has got the grant from the BS, and the SS will use the allocated bandwidth to transmit its own packets. In Fig. 2(b), there is one BE packet arrived in Frame I, and the SS does the contention resolution as mentioned above to send a bandwidth request in Frame I's contention period. But in Frame (I+1), the SS knows that it does not get grant from the BS, the SS will double the contention window size, and transmit bandwidth request again in Frame (I+1)'s contention period, until it gets grant in Frame (I+2), it uses the allocated bandwidth to transmit its own packets.

There is one thing need to be mentioned. If the SS has got the grant from the BS in a frame, the SS cannot transmit bandwidth requests in this frame. This means that if the SS can get grant every time, the SS transmits the bandwidth request once at least two frames (the first frame is for sending the bandwidth request, and the second frame is for sending packets).

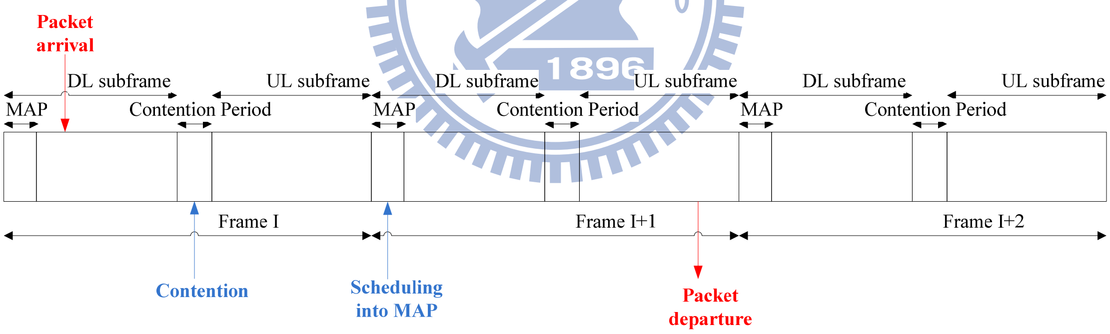

Fig. 2 (a) Type I - Contention mechanism with successfully contend at the 1st time

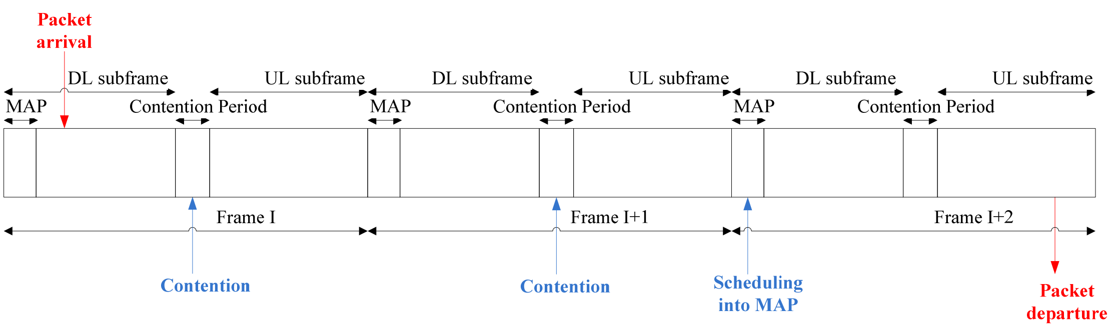

Fig. 2 (b) Type I - Contention mechanism with successfully contend at the 2nd time

2.2 Polling Mechanism

In polling mechanism, the BS polls SSs if they have rtPS, nrtPS, or BE connections. Once the BS polls the SSs, the SSs are allowed to send bandwidth requests to the BS. In the IEEE 802.16 standard [1], it does not give a detailed description to realize the polling mechanism. It does not talk about when the BS should poll the SSs, and we call this period as the polling period. In our method, we put polling periods in the uplink subframes as Fig. 3 shows, just as [5], [14], and [15] do. Here, we only discuss the unicast polling.

As Fig. 3 shows, there is one BE packet arrives in Frame I, and the BS polls this SS in Frame I. The SS will send the bandwidth request at the specified slots which the BS scheduled in the MAP.

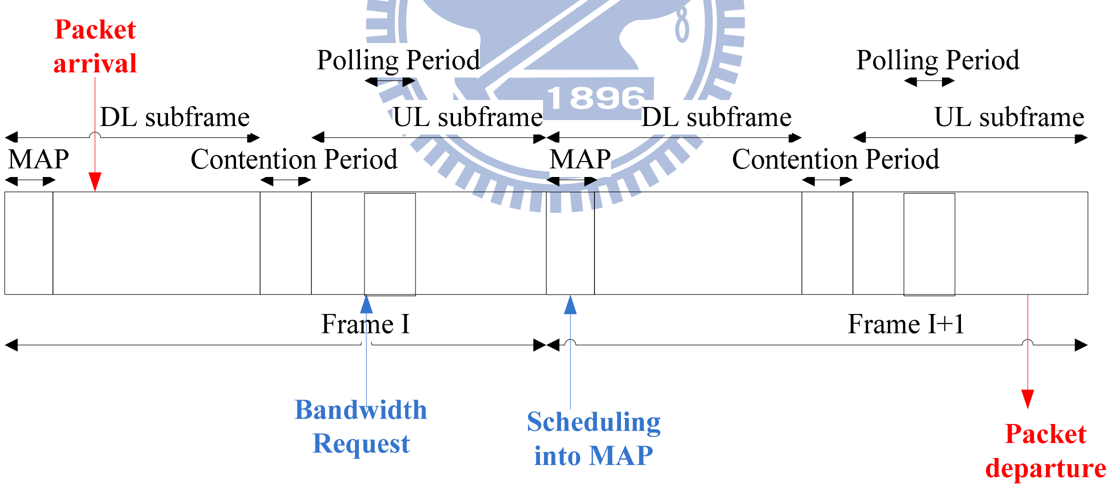

Fig. 3 Polling mechanism

2.3 Piggyback Mechanism

In piggyback mechanism, the SS can send bandwidth request by the piggyback request. If there is one packet can be transmitted in this frame, the SS can request bandwidth for this connection by piggybacking a bandwidth request in the scheduled packet.

As Fig. 4 shows, there is one BE packet arrives in Frame I, and there is one packet can be transmitted in this uplink sub-frame. The SS will add the bandwidth request on the scheduled packet, and transmit the bandwidth request and the scheduled packet to the BS at once.

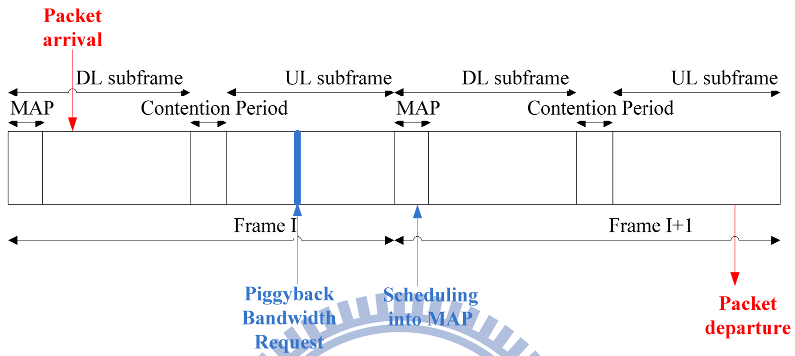

Fig. 4 Piggyback mechanism

Using polling and piggyback mechanisms can prevent from bandwidth requests collision, because they do not need to contend.

Chapter 3. Proposed Bandwidth Request Mechanisms for Best Effort Connections

In the IEEE 802.16 standard [1], BE connections can use contention mechanism, polling mechanism, piggyback mechanism, and bandwidth stealing mechanism to request bandwidth. We have already introduced the Type I - Contention mechanism, polling mechanism, and piggyback mechanism in Chapter 2. In this chapter, we propose two different contention-based and piggyback-based bandwidth request mechanisms, called Type II - Contention mechanism and Modified-Piggyback mechanism to compare with Type I - Contention mechanism and piggyback mechanism illustrated in Chapter 2.

3.1 Type II - Contention Mechanism

Here, we introduce a new contention-based mechanism, called Type II - Contention mechanism, to compare with the Type I - Contention mechanism in section 2.1.

In Type II - Contention mechanism, the operations are similar with Type I - Contention mechanism, the only difference between Type II - Contention mechanism and Type I - Contention mechanism is that once a SS send a bandwidth request to the BS successfully, but the BS does not grant the SS in the following frame, the BS will reserve this bandwidth request for x frames, so the SS should not send bandwidth request in the following x frames. This can reduce the bandwidth request numbers issued by SSs.

In Fig. 5(a), there is one BE packet arrived in Frame I, and the SS does the contention resolution as mentioned above to send a bandwidth request in Frame I's contention period. In Frame (I+1), the SS knows that it has got the grant from the BS, and the SS uses the allocated bandwidth to transmit its own packets. In Fig. 2(b), there is one BE packet arrived in Frame I, and the SS does the contention resolution as mentioned above to send a bandwidth request in Frame I's contention period. In Frame (I+1), the SS knows that it does not get grant from the BS, but because of the Type II - Contention mechanism, the SS will not transmit the bandwidth request again in Frame (I+1)'s contention period, and until it gets the grant in Frame (I+2), it uses the allocated bandwidth to transmit its own packets. If the SS cannot get grant till the Frame (I+*x*-1), the SS will send a new bandwidth request to the BS in Frame (I+*x*).

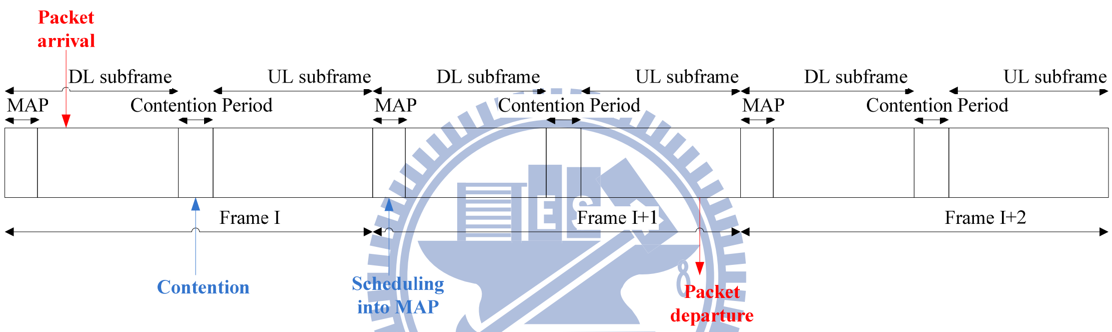

Fig. 5 (a) Type II - Contention mechanism with successfully contend at the 1st time

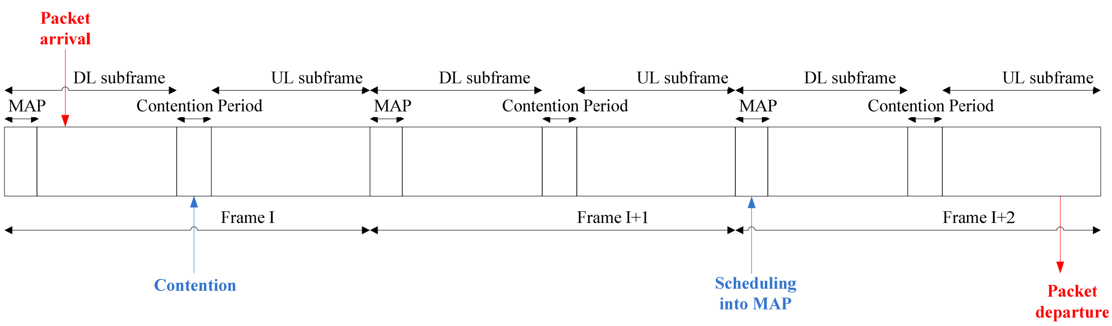

Fig. 5 (b) Type II - Contention mechanism with successfully contend at the 2nd time

3.2 Modified-Piggyback Mechanism (M-Piggyback)

In the IEEE 802.16 standard [1], it does not have many descriptions about the piggyback requests, and in the standard, it just talks that the piggyback request is used for the CID, but it does not say it cannot help other connections to request bandwidth implicitly. By this reason, we introduce a new piggyback mechanism called Modified-Piggyback mechanism (M-Piggyback) to request bandwidth for the other connections. We want to use this M-Piggyback mechanism to raise the successful probability of bandwidth request transmissions, and this would help the SSs to get grant from the BS.

In M-Piggyback mechanism, the SSs can send bandwidth request by piggyback requests. If there is one packet can be transmitted in this frame, the SSs can request bandwidth by piggybacking a bandwidth request in the scheduled packets, the only difference with piggyback mechanism in 2.3 is that the scheduled packet should not be the same connection with the new arrival packets.

As Fig. 4 shows, there is one packet arrives in Frame I, and there is one packet can be transmitted in this uplink sub-frame. The SS adds the bandwidth request on the scheduled packet, and transmits the bandwidth request and the scheduled packet to the BS at once.

For achieving this concept, we need to modify the grant management subheader which is used to store the piggyback request. Because we want to piggyback requests for other connections, we need to add a field into the grant management subheader for indicating the CID of which connection requests the bandwidth. Fig. 6 shows the modified grant management subheader format. The Piggyback Request field should fill with the requested bytes, and the Request CID should fill with the requested connection ID.

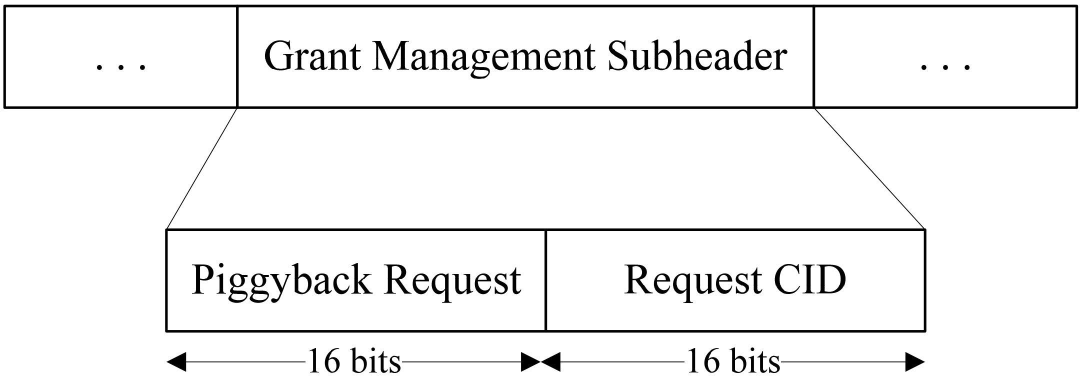

Fig. 6 Modified grant management subheader format

Chapter 4. Performance Evaluation

In this chapter, first we introduce our simulation environment, and then show the simulation results.

4.1 Simulation Environment

Now we know there are many mechanisms for a SS to get grant from the BS. Here we propose 4 scenarios for simulation to observe the differences between those bandwidth request mechanisms in section 4.2.1 – 4.2.4.

1. Contention

The first scenario we call it Contention, in this scenario, all the SSs should send bandwidth request by Type I - Contention mechanism (mentioned at Chapter 2).

2. Polling + Contention

The second scenario we call it Polling + Contention, in this scenario, the BS polls the SSs those have rtPS/nrtPS/BE connections. All the SSs can send bandwidth requests when the BS polls them. If the SSs would not be polled by the BS, they can use Type I - Contention mechanism (mentioned at section 2.1) to contend bandwidth.

3. Polling

The third scenario we call it Polling, in this scenario, BS polls the SSs those have rtPS/nrtPS/BE connections. All the SSs can send bandwidth requests when the BS polls them. If the SSs would not be polled by the BS, they cannot send bandwidth requests to the BS in that frame.

4. Piggyback + Contention

The fourth scenario we call it Piggyback + Contention, in this scenario, SSs with scheduled BE packets could send piggyback requests to the BS (as mentioned at section 2.3). If the SS could not send piggyback requests, it may use Type I - Contention mechanism (mentioned at section 2.1) to contend bandwidth.

In section 4.2.5, we compare the two different contention-based mechanisms, Type I - Contention mechanism and Type II - Contention mechanism. In the Type II - Contention mechanism, the BS will reserve this bandwidth request for *x* frames, and we set x=5 in our simulations.

In section 4.2.6, we compare the two different piggyback-based mechanisms using different scenarios, Piggyback + Contention and M-Piggyback + Contention. The major difference between piggyback mechanism and M-Piggyback mechanism is that the in M-Piggyback mechanism, the SSs can send piggyback requests to the BS with scheduled rtPS, nrtPS, or BE packets.

Table 1. Simulation parameters

Parameters	**Values**
Simulation time	10000 ms
Number of BS	1
Number of SS	10, 25, 50
Frame duration	10 ms
Number of symbols per frame	420
Modulation	QAM64 2/3 (96 bytes/symbol)
Number of physical slots Per frame	28560
Uplink allocation	198 symbols
Downlink allocation	202 symbols
Preambles	10 symbols
Contention period	10 symbols
Packet payload size	174 bytes

In the following, we introduce our simulation environment. We use Visual C++ as the simulation tool. Table 1 shows the simulation parameters.

In our topology, there are 10 to 50 SSs and one BS. There are 420 symbols in one frame, and each frame is 10ms long. Uplink allocation is 198 symbols, downlink allocation is 202 symbols, and the contention period allocation is 10 symbols in one frame. Each symbol has 68 physical slots. Here, we fix the data rate 96 bytes/symbol with modulation QAM64 2/3, and the packet payload with 174bytes. In addition, each simulation executes for 10000ms, and uses error-free channel (this means the packets fail only due to collisions).

Here, we assume there are 8 rtPS and 8 nrtPS connections in our system. rtPS rate is between 512Kbps to 1024Kbps, and nrtPS rate is between 256Kbps to 512Kbps. Each SS has one BE connection with packets generation by exponential inter-arrival time = 10ms.

The BS polls rtPS connections every 3 frames, nrtPS connections every 5 frames, and BE connections every 10 frames.

First, we simulate the 4 scenarios as mentioned above with SS = 25, and the BE packets generation with inter-arrival time = 10ms. Then simulate the 4 scenarios with SS = 50, and the BE packets generation with inter-arrival time = 10ms. For the last, we simulate the 4 scenarios with SS = 10, and the BE packets generation with inter-arrival time = 10ms.

The situation with SS = 25 and 50 means the heavy traffic loads, and SS = 10 means the light traffic loads.

Heavy traffic loads means that there is no enough bandwidth can be allocated for all SSs; light traffic loads means that there is enough bandwidth can be allocated for all SSs.

The following are the metrics which we observed.

1. Average bandwidth request delay: This is the average time for an SS to get grant from the BS, and this is starting from the time that the SS send its bandwidth request to the time that the SS get grant from the BS.
2. Average packet delay (ms): This is the average time for an SS to transmit a BE

packet to the BS, and this is starting from the time that the SS generate the packet to the time that the SS transmit the packet to the BS.

3. Average bandwidth requests per packet: This is the average of total number of bandwidth requests which are requested by all SSs for each packet.
4. % of getting grant: This is the percentage of bandwidth requests is getting from grants for all SSs, and we calculate this value by the equation (1)

$$\frac{\text{Total grant bandwidth}}{\text{Total requested bandwidth}} \times 100\% \tag{1}$$

5. Total amount of bandwidth requests: This is the total number of bandwidth requests which are requested by all SSs.

Fig. 7 shows how to calculate packet delay and bandwidth request delay.

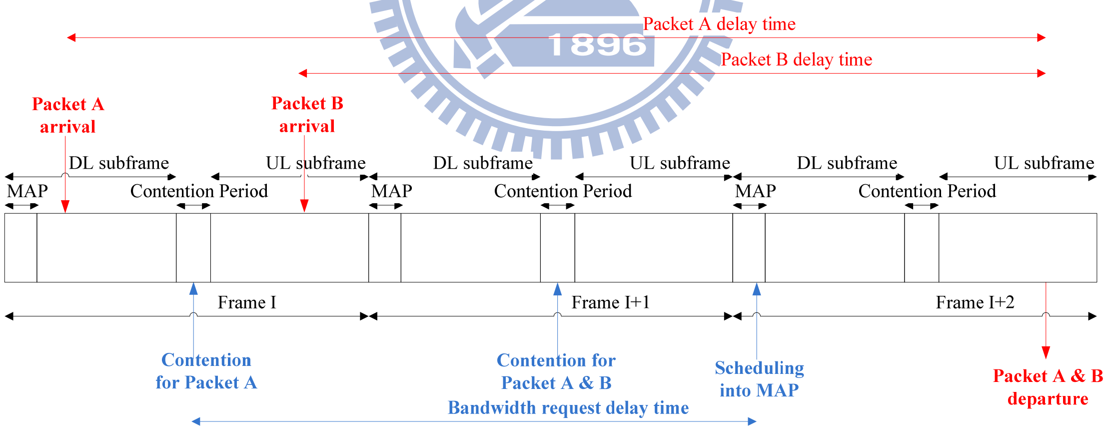

Fig. 7 Illustration of packet delay and bandwidth request delay

4.2 Simulation Results

There are some papers discussing about the analysis of contention period in the IEEE 802.16 WiMAX networks. In these papers, the authors propose equations to determine the length of contention periods. In [11], the authors finally say that the best contention period size is 2*N*-1 (*N* means the number of SSs). Though they determine the best length of contention periods, in heavy traffic loads, the BS still cannot grant the SS because of the inefficient bandwidth (the longer contention period, the shorter uplink subframe). In Fig. 8, we run a simulation using the contention period which can contain 49 bandwidth requests (N=25) as [11] suggests, x-axis is the different scenarios (1:Contention 2:Polling + Contention 3: Polling 4: Piggyback + Contention), and y-axis is the average bandwidth request delays with heavy traffic load (N=25). From the figure, we can know using different bandwidth request mechanisms can reduce the bandwidth request delays more helpfully, so in the following, we discuss bandwidth request mechanisms proposed in Chapter 2.

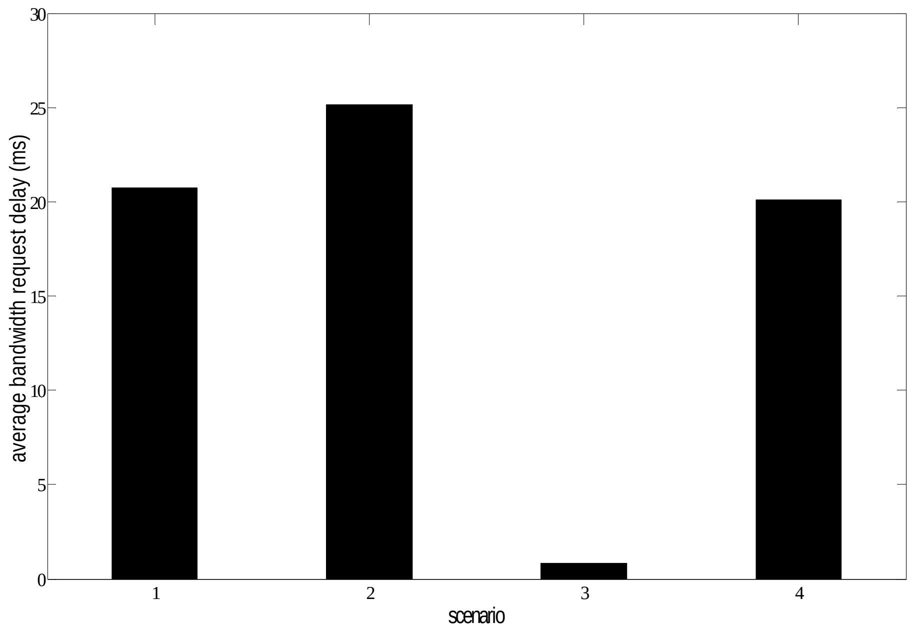

Fig. 8 Average bandwidth request delays upon heavy traffic loads (N=25)

4.2.1 Bandwidth Requests

First, we discuss the amount of average bandwidth requests per packet with different scenarios. In Fig. 9, Fig. 10, and Fig. 11, x-axis is the different scenarios, and y-axis is the average bandwidth requests per packet. In addition, the black bar means the average bandwidth requests which SS needs to send for getting grants from the BS for one packet, and the white bar means the average polling/piggyback bandwidth requests of those bandwidth requests (the value of black bar) which the SS needs to send for getting grants from the BS for one packet.

As Fig. 9, Fig. 10, and Fig. 11 show, Piggyback + Contention (scenario 4) has the largest amount of bandwidth requests in these 4 scenarios. The second one is Contention (scenario 1), and the third one is Polling + Contention (scenario 2). Polling (scenario 3) has the smallest amount of bandwidth requests in these 4 scenarios.

No matter the traffic load is heavy or light, the order of the amount of average bandwidth requests per packet between 4 scenarios is the same (Piggyback + Contention > Contention > Polling + Contention >>> Polling).

First, we should know the intuitive fact that the order of total amount of bandwidth requests sending by SSs decreasingly is Piggyback + Contention > Polling + Contention > Contention >>> Polling. Using Piggyback + Contention sends more average bandwidth requests per packet than Contention because the piggyback mechanism can send more bandwidth requests by the scheduled packets. As the same reason, using Polling + Contention sends more bandwidth requests than Contention, because the BS will poll SSs additionally. In these simulations, each SS will generate about one BE packet every frame, so using Piggyback + Contention will be more bandwidth requests than Polling + Contention. Polling has the smallest amount of bandwidth requests because the SSs need to wait until the BS polls them, the SSs can send bandwidth requests.

Now, we consider about the amount of average bandwidth requests per packet. Using Polling mechanism can send bandwidth requests periodically, but using contention mechanism cannot guarantee the time to get grants, and this means that using Contention may need to send a lot of bandwidth requests for one packet to transmit to the BS than Polling + Contention. Therefore, Contention will send more bandwidth requests in average than Polling + Contention. Finally, we know the order of the amount of average bandwidth requests per packet decreasingly is Piggyback + Contention > Contention > Polling + Contention >>> Polling.

In addition, from Fig. 9, Fig. 10, and Fig. 11, we can find the heavier traffic loads, the lower percentage of bandwidth requests is using polling/piggyback bandwidth requests.

In Fig. 12, x-axis is the different traffic loads, and y-axis is the average bandwidth requests per packet with different scenarios. As Fig. 12 shows, heavier traffic loads would cause more bandwidth requests per packet with different scenarios.

In Fig. 13, we let the packet queues at most can store 100 packets, and x-axis is the different scenarios, and y-axis is the average bandwidth requests per packet upon N=25. From this figure, we can find the only difference between limited size packet queues and unlimited size packet queues is that Polling + Contention has fewer bandwidth requests than Polling. This is because the requested bytes become lesser than the unlimited size packet queues, and lesser requested bytes cause higher grant probabilities.

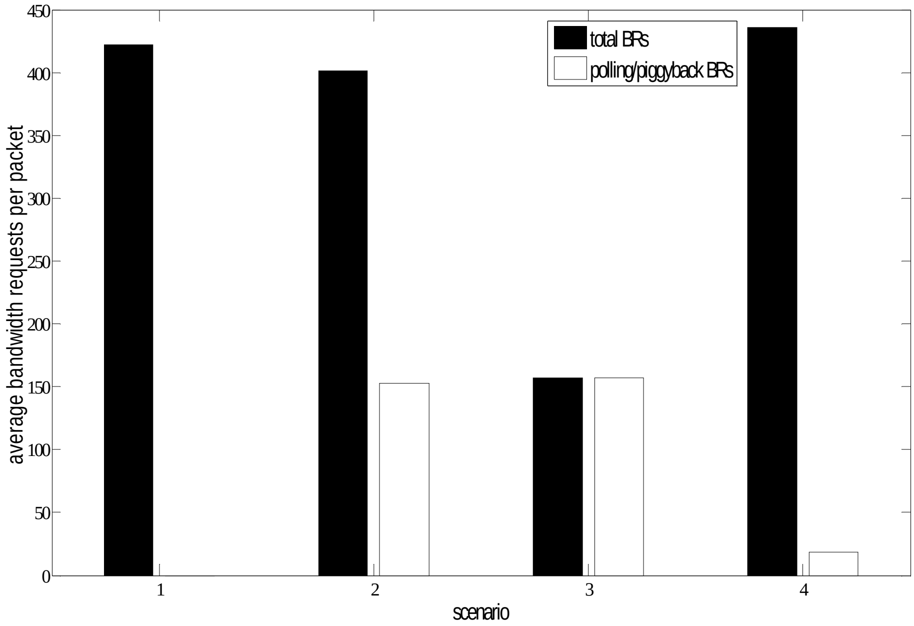

Fig. 9 Average bandwidth requests per packet with heavy traffic loads (N=25)

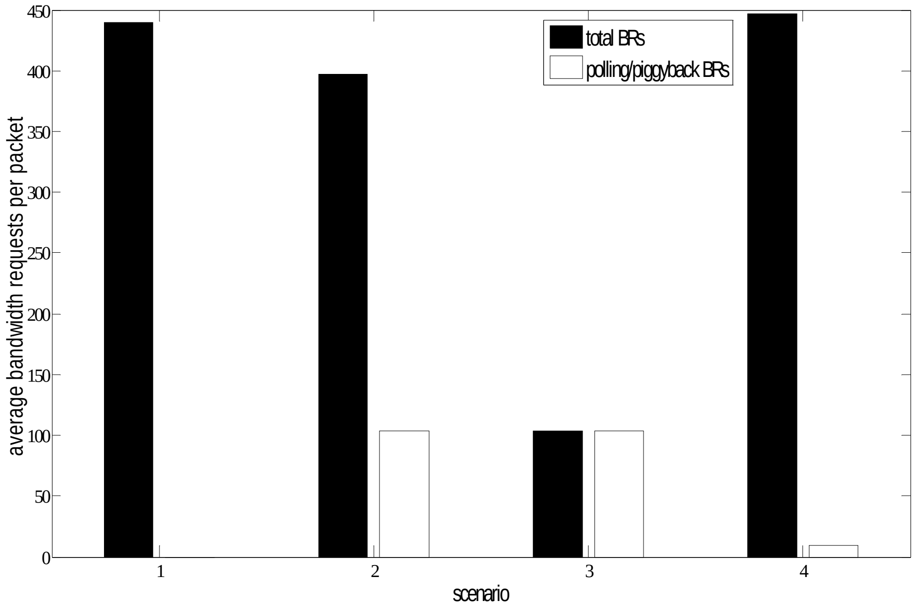

Fig. 10 Average bandwidth requests per packet with heavy traffic loads (N=50)

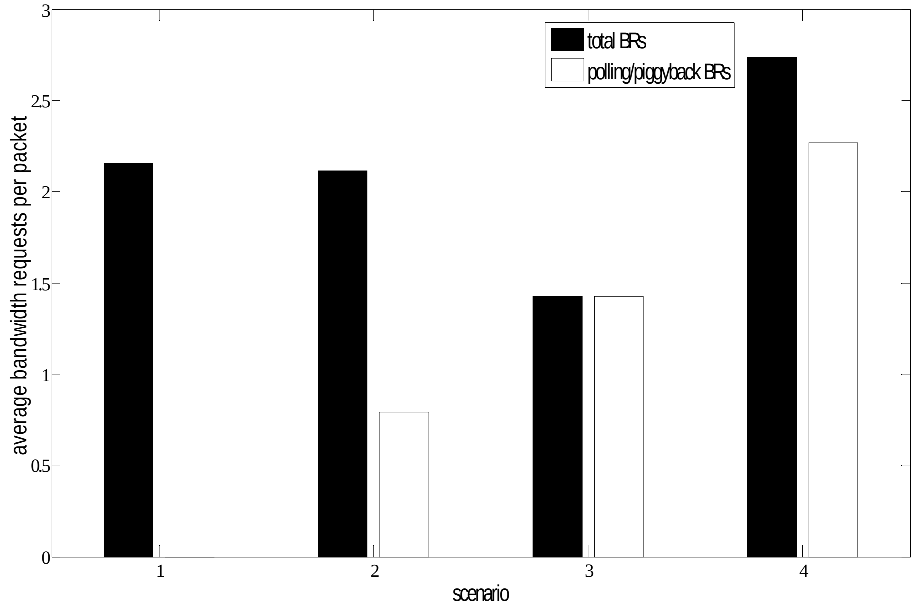

Fig. 11 Average bandwidth requests per packet with light traffic loads (N=10)

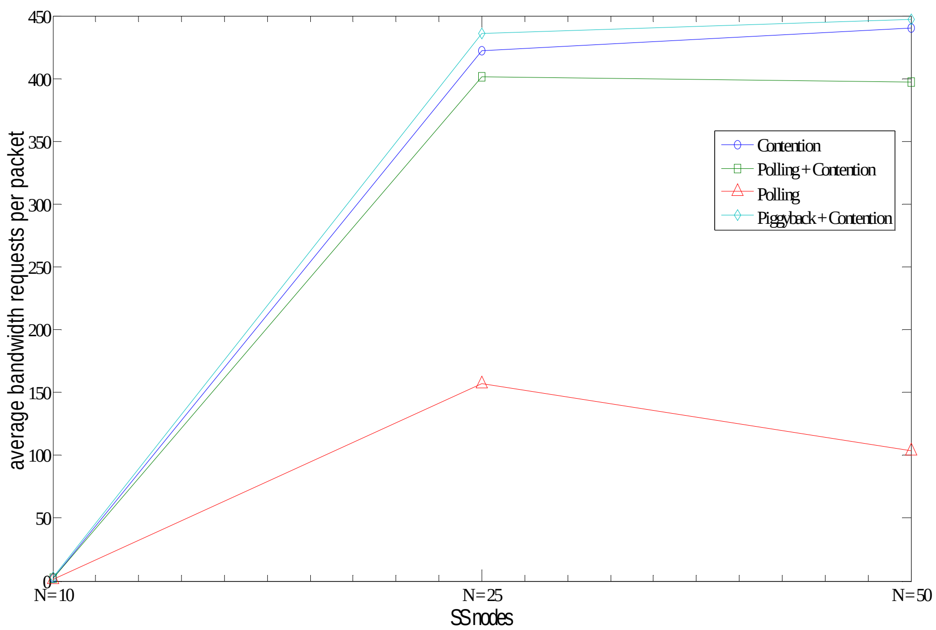

Fig. 12 Average bandwidth requests per packet with different traffic loads

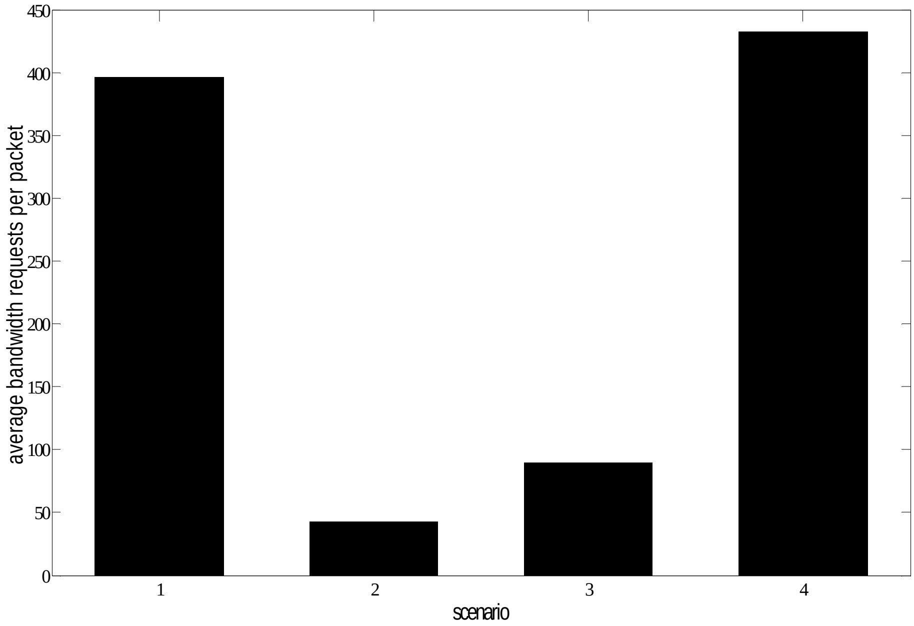

Fig. 13 Average bandwidth requests per packet with limited buffer size upon N=25

4.2.2 Percentage of Getting Grants

Next, let us discuss the percentage of getting grant with different scenarios. In Fig. 14 and Fig. 15, x-axis is the different scenarios, and y-axis is the percentage of getting grant. In addition, black bar means the value with N=50, and white bar means the value with N=25 in Fig.8 with heavy traffic load; black bar means the value with N=10 in Fig. 15 with light traffic load.

Fig. 14 shows the percentage of getting grant with heavy traffic loads. Polling (scenario 3) has the highest percentage of getting grant in these 4 scenarios. The following are Contention (scenario 1) and Piggyback + Contention (scenario 4). The last is Polling + Contention (scenario 2).

Using Polling gets the bandwidth easily because of the lowest collision probability in these 4 scenarios. As mentioned at section 2.2, polling mechanism will not cause any collision. Contention and Piggyback + Contention have higher collision probabilities than Polling, so they get bandwidth harder from the BS than Polling. Although Polling + Contention has lower collision probability than Contention, Polling + Contention still gets bandwidth harder than Contention. This is because the heavy traffic loads. Heavy traffic loads lead the BS to discard many bandwidth requests, and Polling + Contention sends more bandwidth requests than Contention, at this situation, in Polling + Contention, BS needs to discard more bandwidth requests than Contention. Finally, Contention can have higher grant probability than Polling + Contention.

Here, we find that Piggyback + Contention performs like Contention. This is also because of the heavy traffic loads. In heavy traffic loads, the bandwidth requests sending by piggyback requests is few in Piggyback + Contention, and this means the most bandwidth requests are sending by the contention mechanism. So, Piggyback + Contention performs like Contention.

Fig. 15 shows the percentage of getting grant with light traffic loads. Polling (scenario 3) has the highest percentage of getting grant in these 4 scenarios. The following is Piggyback + Contention (scenario 4), and the third one is Polling + Contention (scenario 2). The last one is Contention (scenario 1).

Using Polling gets the bandwidth easily because of the lowest collision probability in these 4 scenarios. As mentioned before, polling mechanism will not cause any collision. Piggyback + Contention has higher collision probability than Polling, so it gets bandwidth harder from the BS than Polling. But in fact, Piggyback + Contention in light traffic loads has low collision probability because of the high percentage of bandwidth requests sending by piggyback requests. Though the percentage of bandwidth requests sending by polling mechanism is not low, Polling + Contention still has higher collision probability than Piggyback + Contention because of the lower percentage of polling BRs/piggyback BRs. So Polling + Contention gets bandwidth hardly than Piggyback + Contention from the BS. Contention has the highest collision probability, so it gets bandwidth hardest among these 4 scenarios.

In Fig. 16, x-axis is the different traffic loads, and y-axis is the percentage of getting grant with different scenarios. As Fig. 16 shows, heavier traffic loads would decrease the percentage of getting grant with different scenarios.

In Fig. 17, we let the packet queues at most can store 100 packets, and x-axis is the different scenarios, and y-axis is the percentage of getting grant upon N=25. From this figure, we can find that with limited size packet queues, the percentage of getting grant is higher than with unlimited size packet queues. This is because the requested bytes become lesser than the unlimited size packet queues, and lesser requested bytes cause higher grant probabilities.

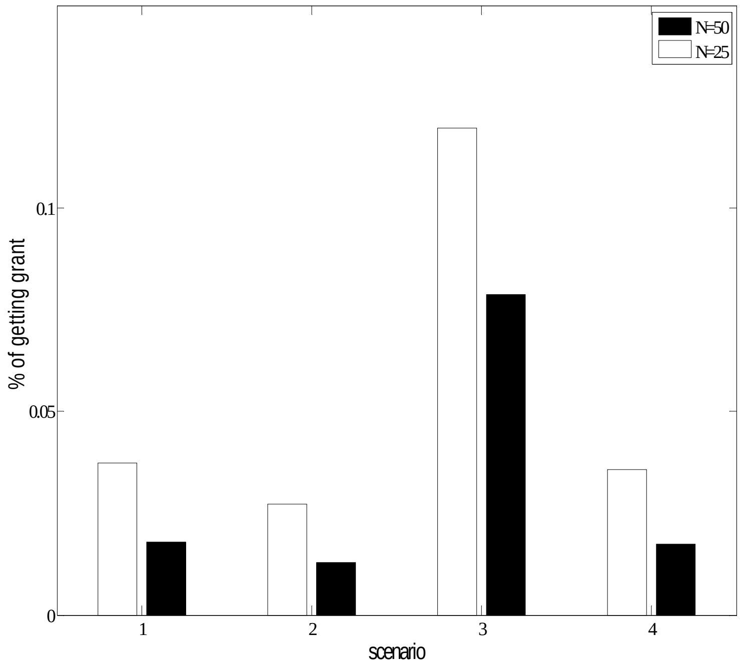

Fig. 14 Percentage of getting grant with heavy traffic loads

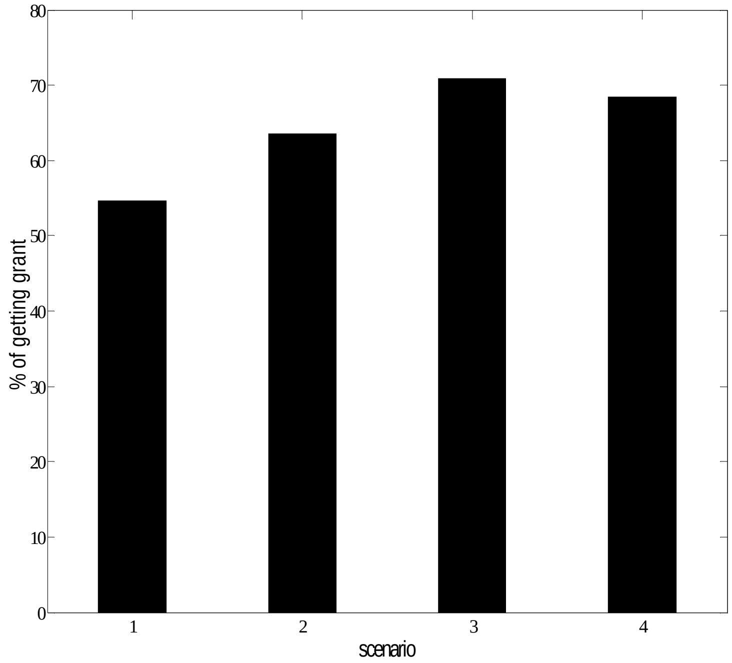

Fig. 15 Percentage of getting grant with light traffic loads upon N=10

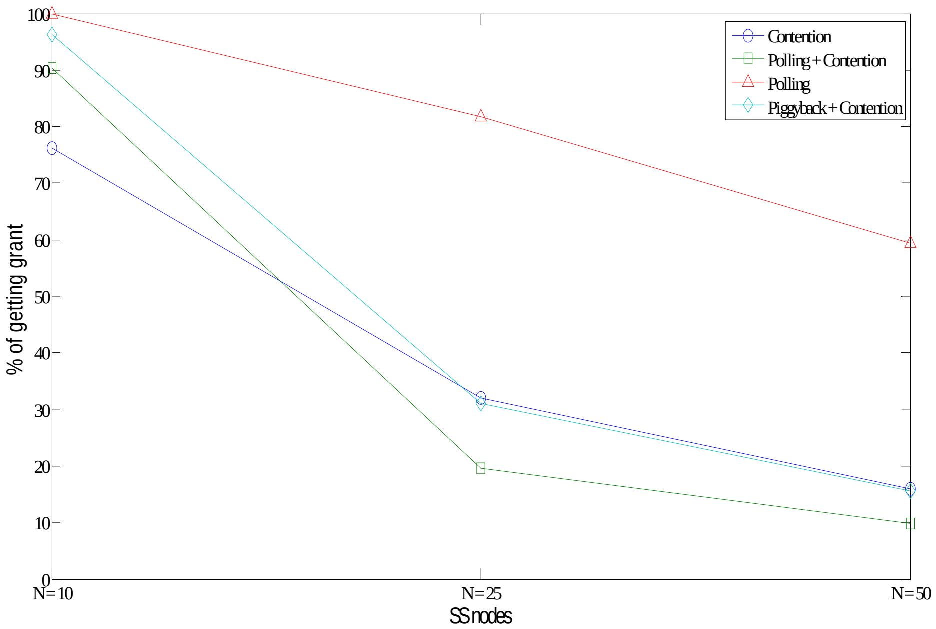

Fig. 16 Percentage of getting grant with different traffic loads

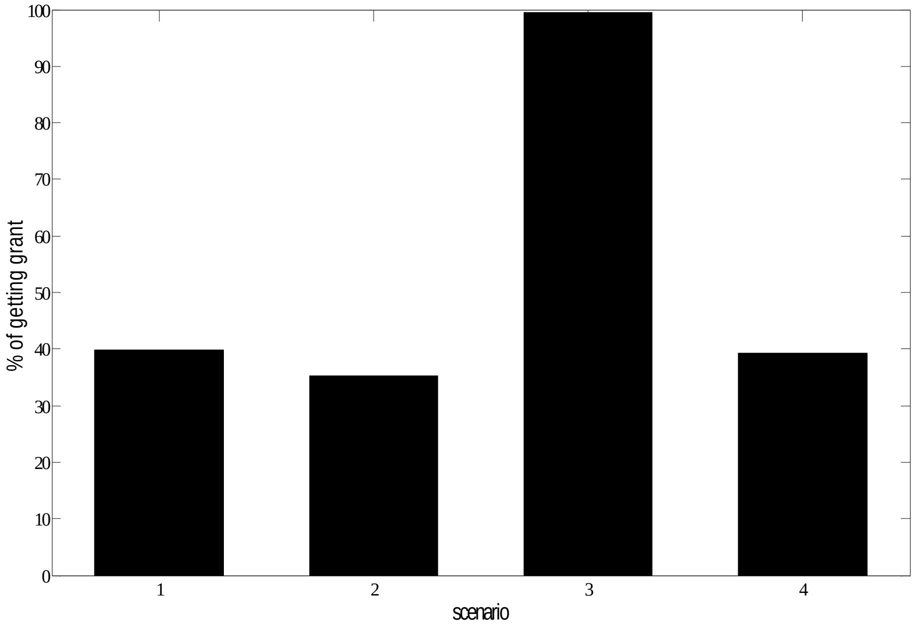

Fig. 17 Percentage of getting grant with limited buffer size upon N=25

4.2.3 Average Bandwidth Request Delays

Then, let us discuss the average bandwidth request delays with different scenarios. In Fig. 18 and Fig. 19, x-axis is the different scenarios, and y-axis is the average bandwidth request delays. In addition, black bar means the value with N=50, and white bar means the value with N=25 in Fig.10 with heavy traffic load; black bar means the value with N=10 in Fig. 19 with light traffic load.

Fig. 18 shows the average bandwidth request delays with heavy traffic loads. Polling + Contention (scenario 2) has the longest bandwidth request delay in these 4 scenarios. The following are Piggyback + Contention (scenario 4). The third one is Contention (scenario 1). Polling (scenario 3) has the shortest bandwidth request delay in these 4 scenarios.

The order of average bandwidth request delays between these 4 scenarios is related to the percentage of getting grant. The higher percentage of getting grant is, the shorter average bandwidth request delay is.

Fig. 19 shows the average bandwidth request delays with light traffic loads, and all the average bandwidth request delays are very short (shorter than one frame duration). In addition, they also related to the percentage of getting grant.

In Fig. 20, x-axis is the different traffic loads, and y-axis is the average bandwidth request delays with different scenarios. As Fig. 20 shows, heavier traffic loads would increase the bandwidth request delays with different scenarios.

In Fig. 21, we let the packet queues at most can store 100 packets, and x-axis is the different scenarios, and y-axis is the average bandwidth request delays upon N=25. From this figure, we can find that with limited size packet queues, the bandwidth request delays are higher than with unlimited size packet queues. This is because the requested bytes become lesser than the unlimited size packet queues, and lesser requested bytes cause higher grant probabilities.

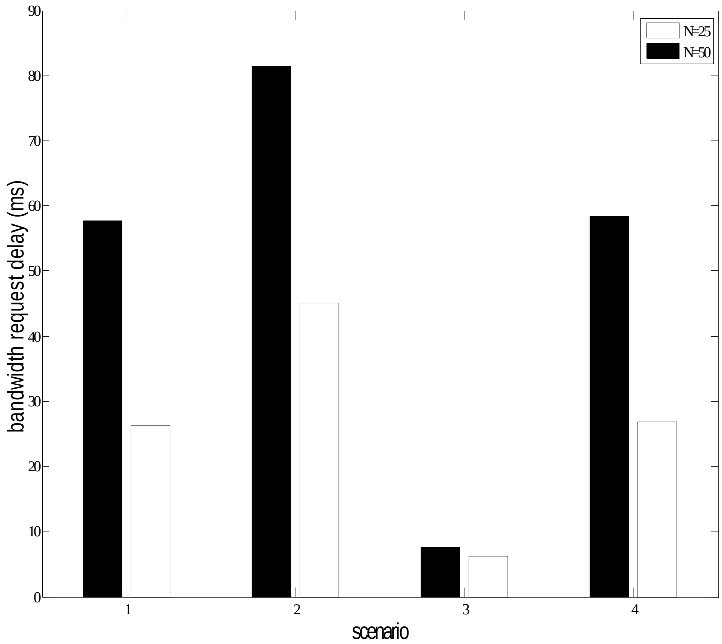

Fig. 18 Average bandwidth request delays with heavy traffic loads

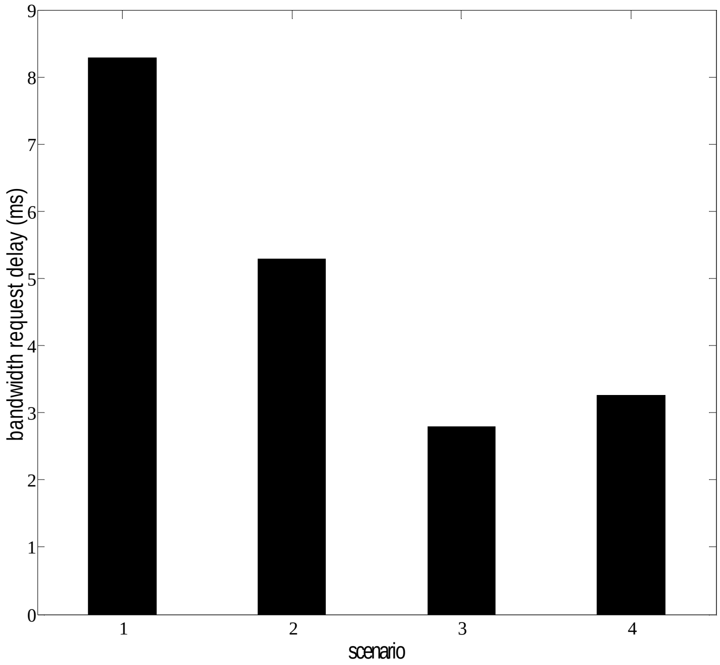

Fig. 19 Average bandwidth request delays with light traffic loads upon N=10

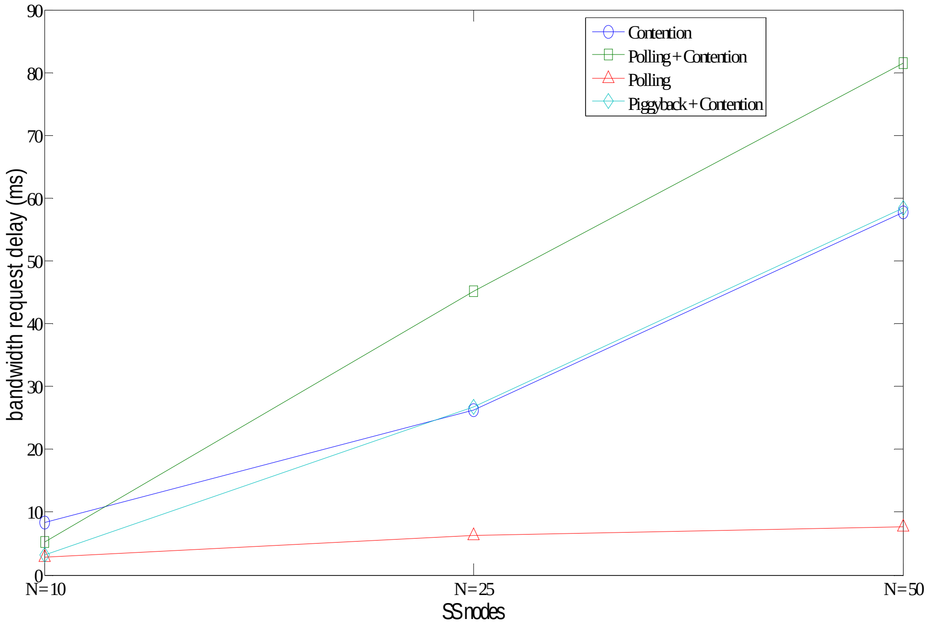

Fig. 20 Average bandwidth request delays with different traffic loads

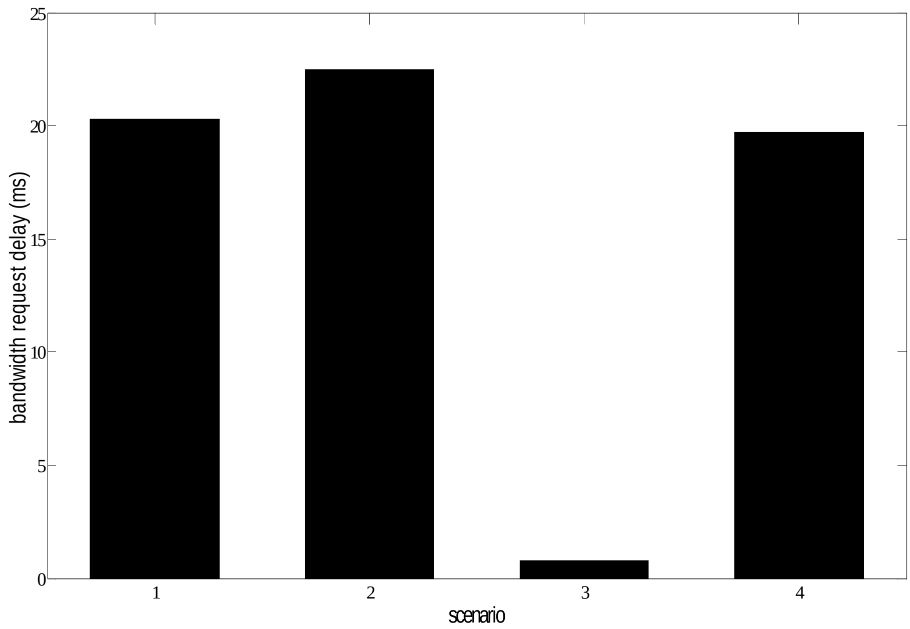

Fig. 21 Average bandwidth request delays with limited buffer size upon N=25

4.2.4 Average Packet Delays

Then, we discuss the average BE packet delays with different scenarios. In Fig. 22 and Fig. 23, x-axis is the different scenarios, and y-axis is the average packet delays. In addition, black bar means the value with N=50, and white bar means the value with N=25 in Fig.12 with heavy traffic load; black bar means the value with N=10 in Fig. 23 with light traffic load.

Fig. 22 shows the average packet delays with heavy traffic loads. Contention (scenario 1) and Piggyback + Contention (scenario 4) have the longest average packet delay in these 4 scenarios. Next is Polling + Contention (scenario 2). The Least one in these 4 scenarios is Polling (scenario 3).

Both of Polling and Polling + Contention are using polling mechanism to send bandwidth requests. Using polling mechanism can send bandwidth requests periodically, and has higher percentage of getting grant, but using contention mechanism cannot guarantee the time to get grants. So Contention is worse than Polling and Polling + Contention. In addition, Piggyback + Contention performs like Contention because of the low percentage of piggyback BRs as mentioned before.

Fig. 23 shows the average packet delays with light traffic loads. Polling (scenario 3) has the longest average packet delay in these 4 scenarios. The second one is Contention (scenario 1), and the third one is Polling + Contention (scenario 2). The least one is Piggyback + Contention (scenario 4).

Polling has the longest average packet delay because it needs to wait for the BS polls the SSs. The other 3 scenarios are related to the percentage of getting grant. The higher percentage of getting grant is, the shorter average packet delay is.

In Fig. 24, x-axis is the different traffic loads, and y-axis is the average packet delays with different scenarios. As Fig. 24 shows, heavier traffic loads would cause longer packet delays with different scenarios.

In Fig. 25, we let the packet queues at most can store 100 packets, and x-axis is the different scenarios, and y-axis is the average packet delays upon N=25. From this figure, we can find the only difference between limited size packet queues and unlimited size packet queues is that the packet delay of Polling + Contention is shorter than Polling. This is because the requested bytes become lesser than the unlimited size packet queues, lesser requested bytes cause higher grant probabilities, and higher grant probabilities cause shorter packet delays.

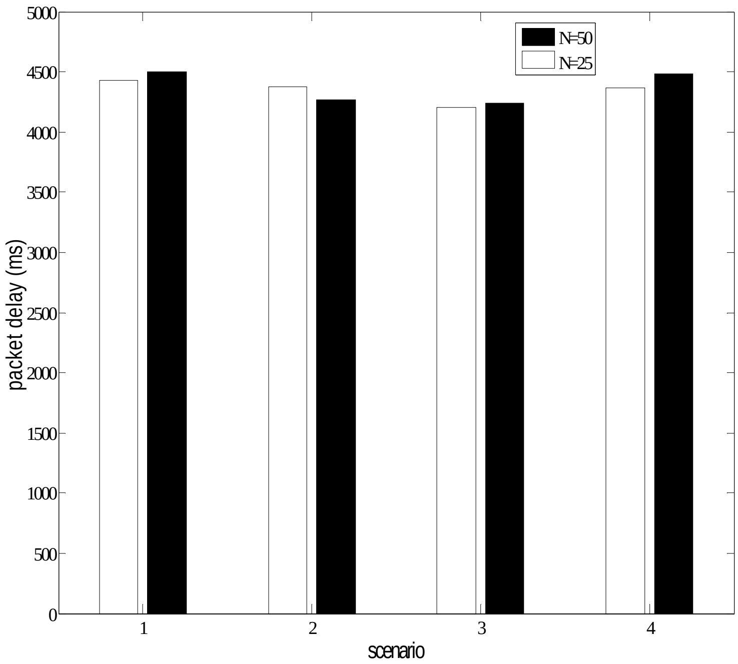

Fig. 22 Average packet delays with heavy traffic loads

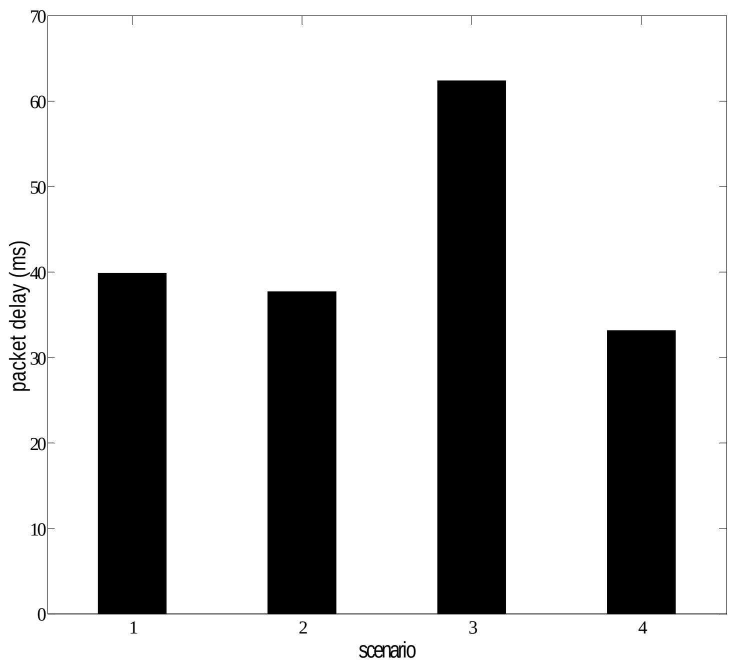

Fig. 23 Average packet delays with light traffic loads upon N=10

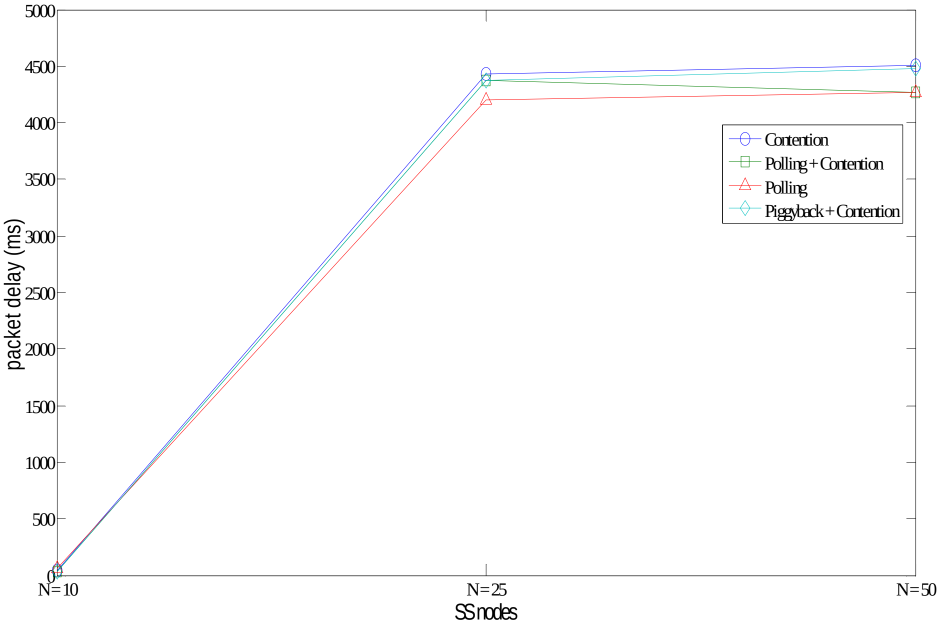

Fig. 24 Average packet delays with different traffic loads

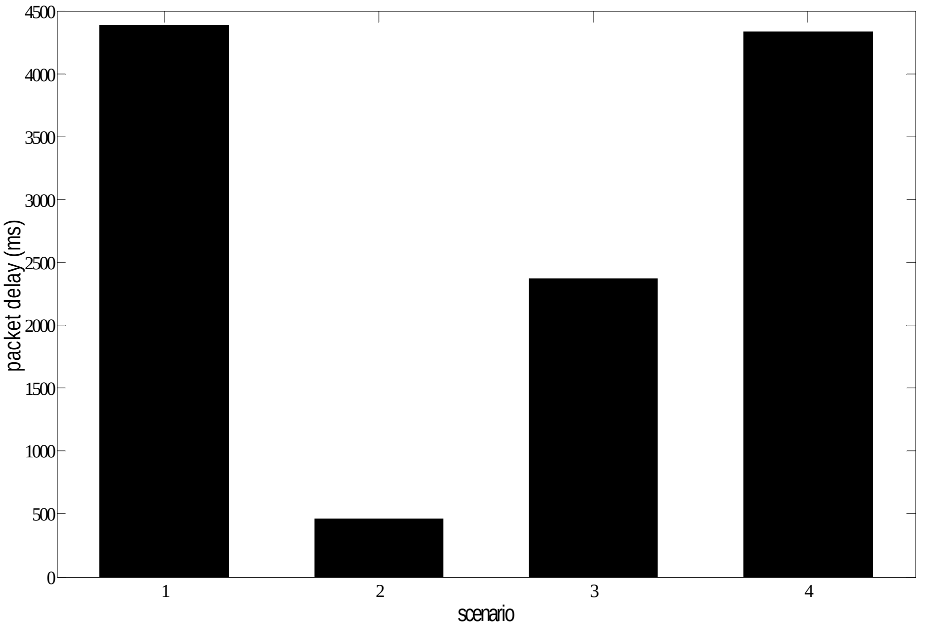

Fig. 25 Average packet delays with limited buffer size upon N=25

4.2.5 Type I - Contention vs. Type II - Contention Mechanisms

In this part, we want to discuss the difference between Type I - Contention mechanism and Type II - Contention mechanism. Here, we think about the following metric, the total amount of bandwidth requests sending by all SSs with the two different contention-based bandwidth request mechanisms.

Let us discuss the total amount of bandwidth requests with different scenarios. In Fig. 26, and Fig. 27, x-axis is the different bandwidth request mechanisms, the left is Type I - Contention mechanism; the right is Type II - Contention mechanism, and y-axis is the total amount of bandwidth requests. In addition, the black bar means the total amount of bandwidth requests which SS needs to send for getting grants from the BS with N=50, and the white bar means the total amount of bandwidth requests which SS needs to send for getting grants from the BS with N=25 in Fig.14 with heavy traffic load; black bar means the total amount of bandwidth requests which SS needs to send for getting grants from the BS with N=10 in Fig. 27 with light traffic load.

Fig. 26 shows the total amount of bandwidth requests with heavy traffic loads. From Fig. 26, we can find that in heavy traffic loads, using Type II - Contention mechanism can reduce the number of bandwidth requests sending by the SSs obviously. In addition, as the number of SSs increases, the bandwidth requests will reduce more proportionally, and this means that the higher collision probability, the more reduction of bandwidth requests.

Fig. 27 shows the total amount of bandwidth requests with light traffic loads. From Fig. 27, we can find that in light traffic loads, using Type II - Contention mechanism does not have any difference with Type I - Contention mechanism because of the lower collision probability. So Type II - Contention mechanism does not have effect in light traffic loads.

Now, let us think about the impacts on different packet error rates using Type II - Contention mechanism. In Fig. 28, x-axis is the different bandwidth request packet error rates, and y-axis is the average bandwidth request delays. The left one is using packet error rate = 0 (will not cause any error), the middle one is using packet error rate = 0.1, and the right one is using packet error rate = 0.3.

As Fig. 28 shows, higher packet error rates would increase the bandwidth request delays, and this is because in Type II – Contention mechanism, if the bandwidth requests sending by SSs are lost, the SSs would not know the transmissions are failed and they always think their transmissions are succeed. In this situation, the bandwidth request delays will become longer.

In Fig. 29, x-axis is the different bandwidth request packet error rates, and y-axis is the average packet delays. The left one is using packet error rate = 0 (will not cause any error), the middle one is using packet error rate = 0.1, and the right one is using packet error rate = 0.3.

As Fig. 29 shows, higher packet error rates would increase the packet delays, and this is because in Type II – Contention mechanism, if the bandwidth requests sending by SSs are lost, the SSs would not know the transmissions are failed and they always think their transmissions are succeed. In this situation, same as the impact on bandwidth request delays, the packet delays will become longer, too.

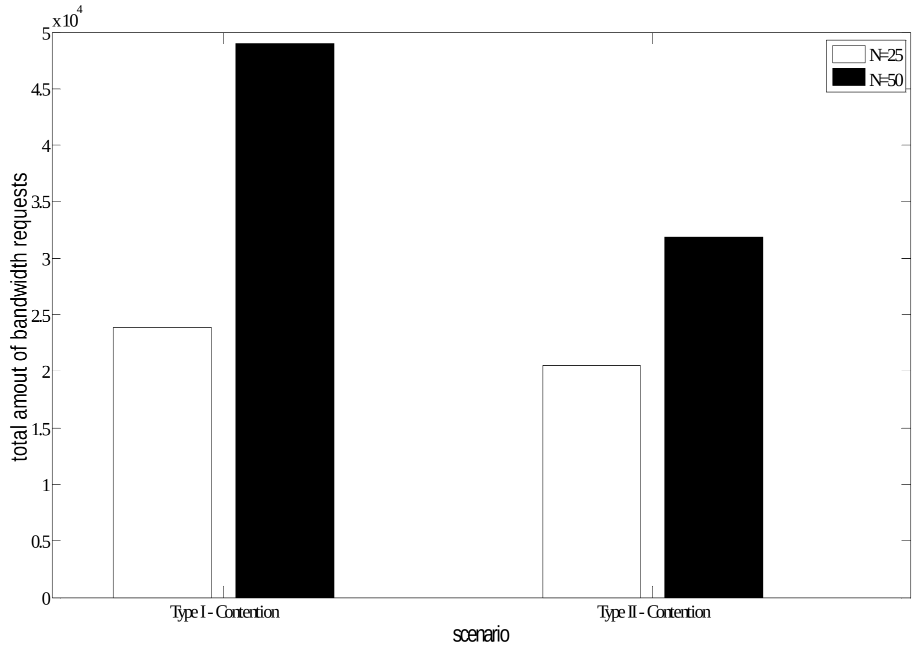

Fig. 26 Total amount of bandwidth requests with heavy traffic loads in different contention-based bandwidth request mechanisms

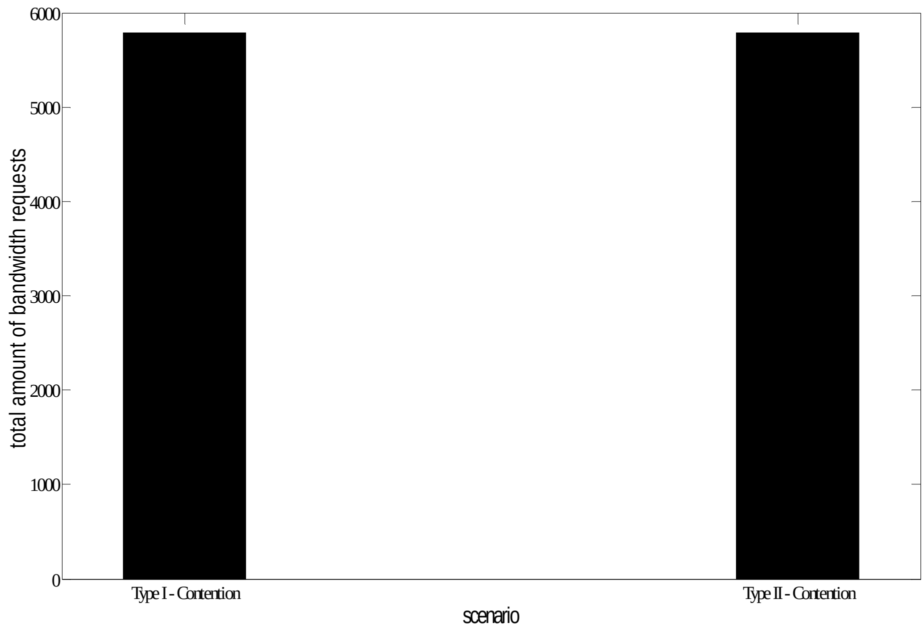

Fig. 27 Total amount of bandwidth requests with light traffic loads in different contention-based bandwidth request mechanisms upon N=10

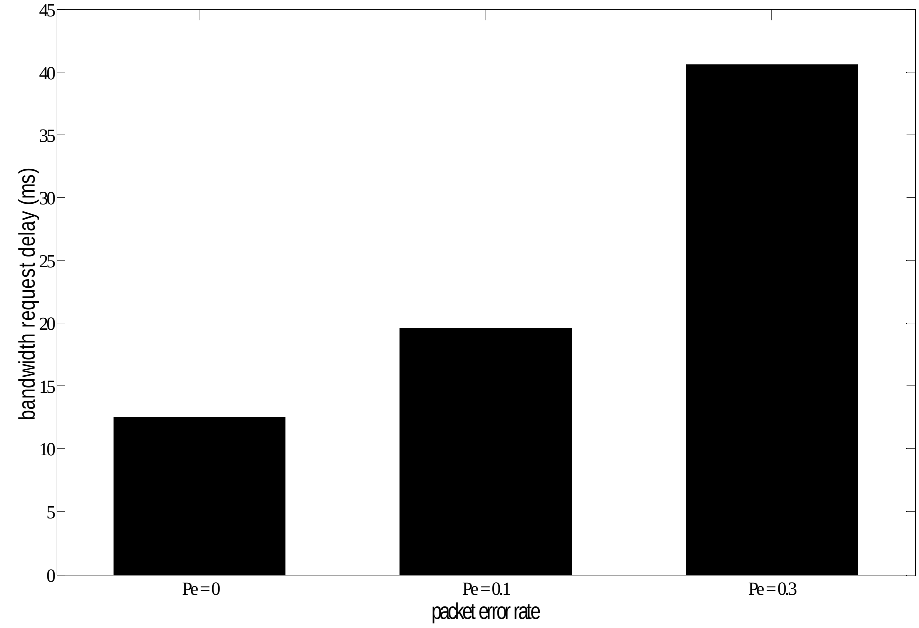

Fig. 28 Average bandwidth request delays with different packet error rates using Type II - Contention mechanism upon N=25

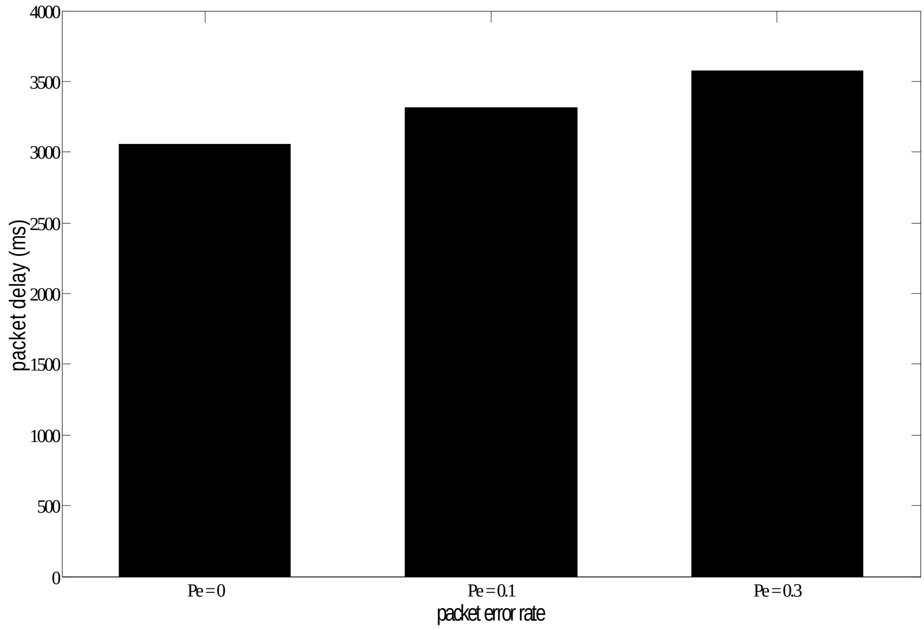

Fig. 29 Average packet delays with different packet error rates using Type II - Contention mechanism upon N=25

4.2.6 Piggyback vs. Modified-Piggyback Mechanisms

In this part, we want to discuss the difference between piggyback mechanism and M-Piggyback mechanism using two different scenarios, Piggyback + Contention and M-Piggyback + Contention. Here, we think about the following metrics, the average bandwidth request delays with the two different piggyback-based bandwidth request mechanisms, and the average packet delays with the two different piggyback-based bandwidth request mechanisms.

First, let us consider the average bandwidth request delays. In Fig. 30, and Fig. 31, x-axis is the different scenarios, and y-axis is the average bandwidth request delays. In addition, black bar means the average bandwidth request delay with N=50, and white bar means the average bandwidth request delay with N=25 in Fig.16 with heavy traffic load; black bar means the average bandwidth request delay with N=10 in Fig. 31 with light traffic load.

Fig. 30 shows the average bandwidth request delays with heavy traffic loads. From Fig. 30, we can find that in heavy traffic loads, using M-Piggyback + Contention scenario can reduce the average bandwidth request delays obviously, and this is related to the average bandwidth requests per packet shows in Fig. 32. This is because the M-Piggyback + Contention can piggyback more requests than Piggyback + Contention, so the average bandwidth requests per packet in M-Piggyback + Contention is lower than Piggyback + Contention.

Fig. 31 shows the average bandwidth request delays with light traffic loads. From Fig. 31, we can find that in light traffic loads, using M-Piggyback + Contention does not have differences with Piggyback + Contention because of the similar piggyback request numbers. So M-Piggyback + Contention does not have effect in light traffic loads.

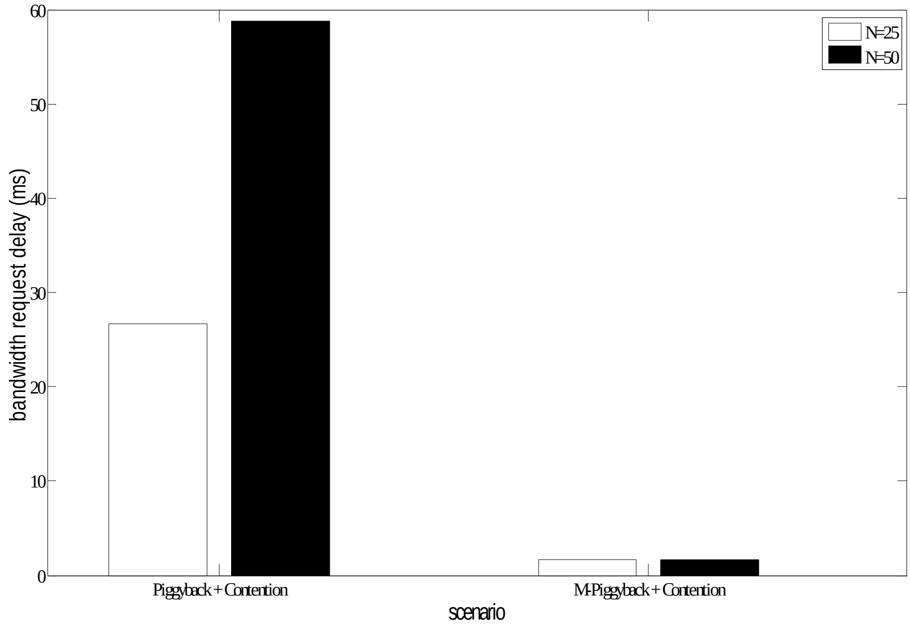

Fig. 30 Average bandwidth request delays with heavy traffic loads in different piggyback-based bandwidth request mechanisms

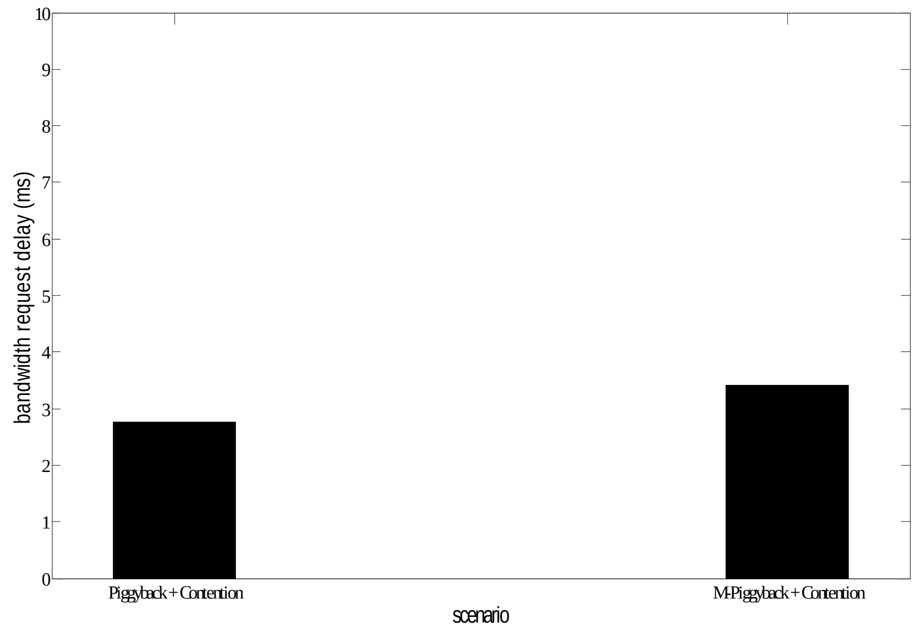

Fig. 31 Average bandwidth request delays with light traffic loads in different piggyback-based bandwidth request mechanisms upon N=10

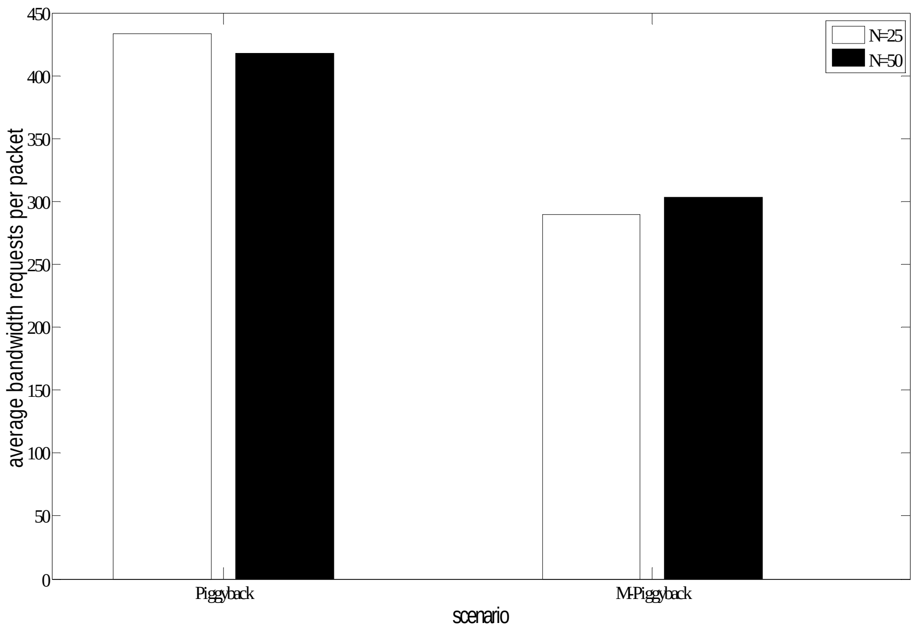

Fig. 32 Average bandwidth requests per packet with heavy traffic loads in different piggyback-based bandwidth request mechanisms

Then, let us consider the average BE packet delays. In Fig. 33, and Fig. 34, x-axis is the different scenarios, and y-axis is the average bandwidth request delays. In addition, black bar means the average packet delay with N=50, and white bar means the average packet delay with N=25 in Fig.21 with heavy traffic load; black bar means the average packet delay with N=10 in Fig. 34 with light traffic load.

Fig. 33 shows the average packet delays with heavy traffic loads. From Fig. 33, we can find that in heavy traffic loads, using M-Piggyback + Contention scenario can reduce the average packet delays obviously, and this is because the M-Piggyback + Contention can piggyback more requests than Piggyback + Contention, so the average packet delay in M-Piggyback + Contention is lower than Piggyback + Contention.

Fig. 34 shows the average packet delays with light traffic loads. From Fig. 34, we can find that in light traffic loads, using M-Piggyback + Contention does not have differences with Piggyback + Contention because of the similar piggyback request numbers. So M-Piggyback + Contention does not have effect in light traffic loads.

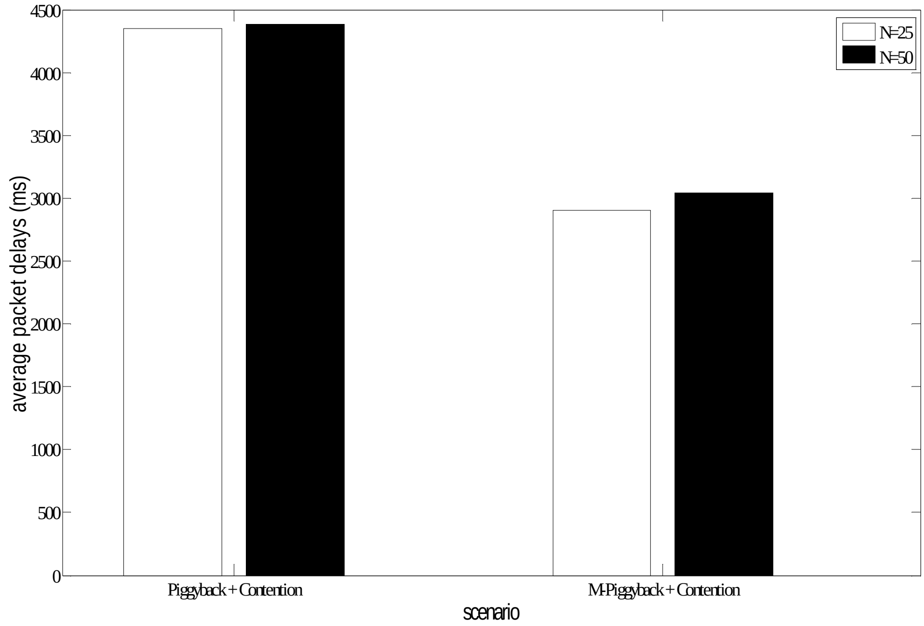

Fig. 33 Average packet delays with heavy traffic loads in different piggyback-based bandwidth request mechanisms

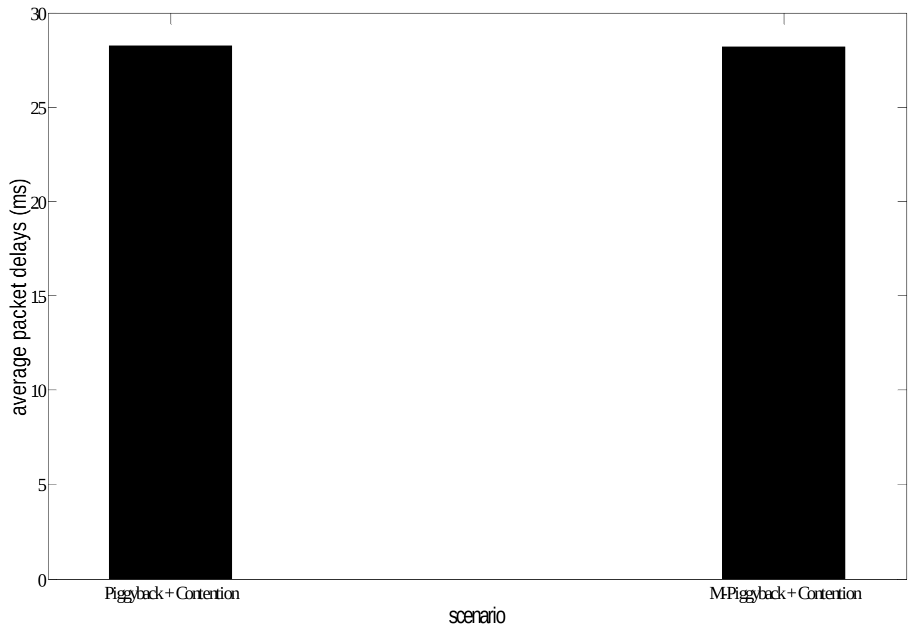

Fig. 34 Average packet delays with light traffic loads in different piggyback-based bandwidth request mechanisms upon N=10

Now, let us concern about the impacts on rtPS and nrtPS connections of the different piggyback-based mechanisms. In Fig. 35, x-axis is the different scenarios, and y-axis is the average rtPS packet delays; in Fig. 36, x-axis is the different scenarios, and y-axis is the throughput of rtPS connections; in Fig. 37, x-axis is the different scenarios, and y-axis is the throughput of nrtPS connections.

From the following three figures, we can know M-Piggyback mechanism will not change the delay and throughputs in rtPS and nrtPS connections. rtPS and nrtPS connections perform like using piggyback mechanism.

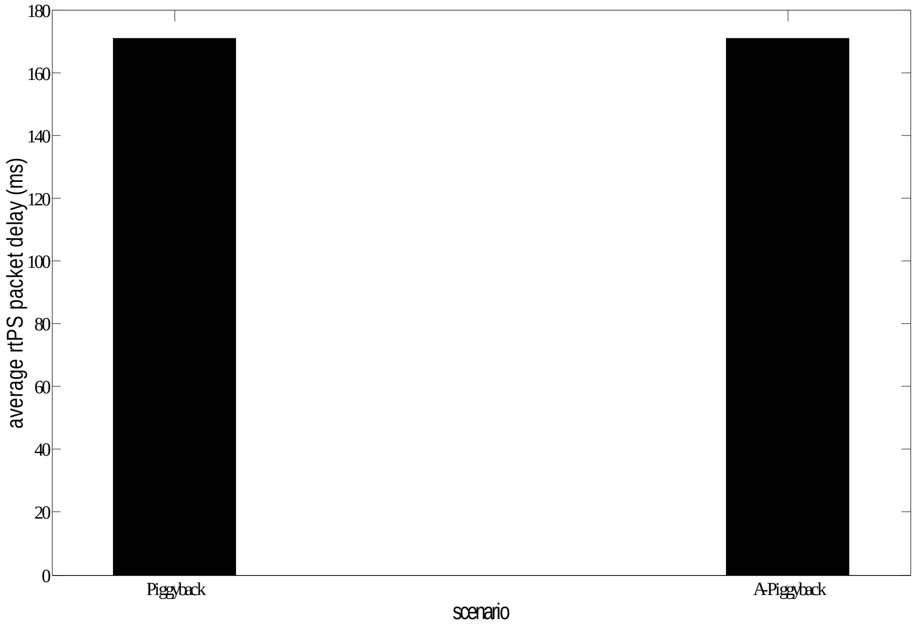

Fig. 35 Average rtPS packet delays with heavy traffic loads in different piggyback-based bandwidth request mechanisms

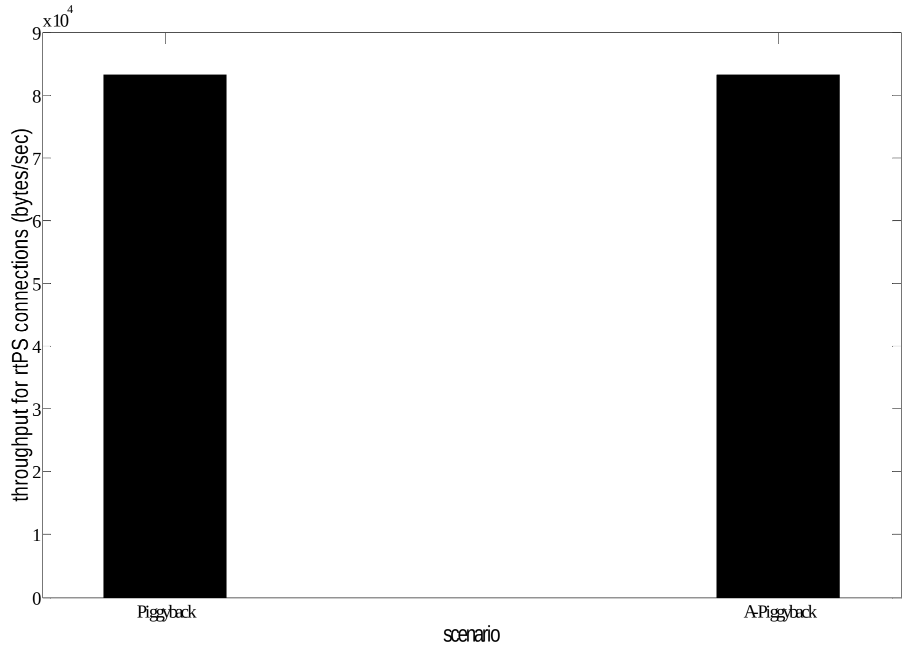

Fig. 36 Throughput of rtPS connections with heavy traffic loads in different piggyback-based bandwidth request mechanisms

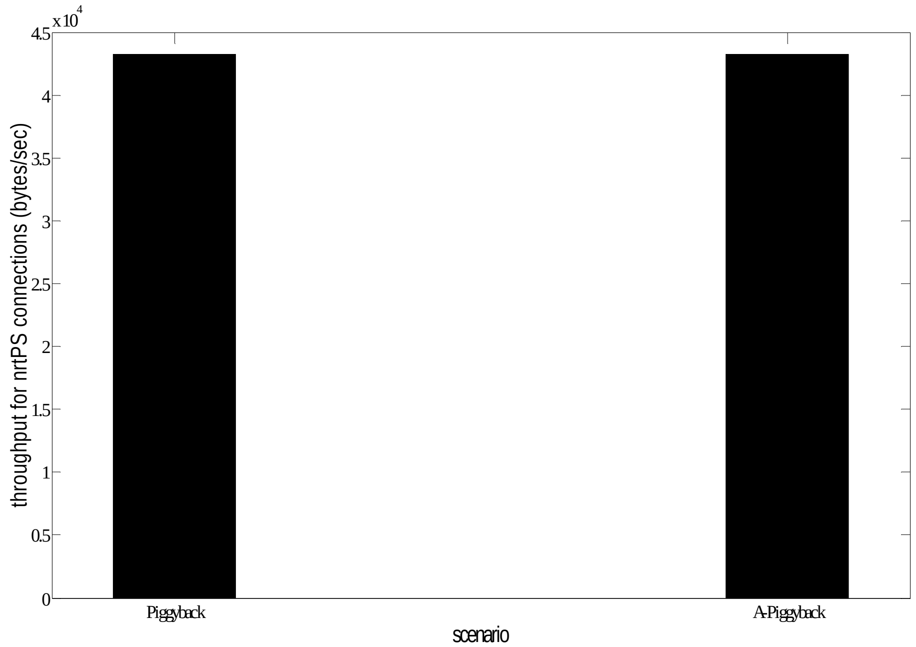

Fig. 37 Throughput of nrtPS connections with heavy traffic loads in different piggyback-based bandwidth request mechanisms

4.2.7 Summary

We have discussed the four parameters in our simulation, and they are average bandwidth requests per packet, percentage of getting grant, average bandwidth request delays, and average packet delays.

In heavy traffic load situation, we can find the following relationships.

1. Average bandwidth requests per packet

   Piggyback + Contention > Contention > Polling + Contention >> Polling

2. Percentage of getting grant

   Polling >> Contention, Piggyback + Contention > Polling + Contention

3. Average bandwidth request delays

   Polling + Contention > Piggyback + Contention, Contention >> Polling

4. Average packet delays

   Contention, Piggyback + Contention > Polling + Contention > Polling

In light traffic load situation, we can find the following relationships.

1. Average bandwidth requests per packet

   Piggyback + Contention > Contention > Polling + Contention > Polling

2. Percentage of getting grant

   Polling >> Piggyback + Contention > Polling + Contention > Contention

3. Average bandwidth request delays

   Contention > Polling + Contention > Piggyback + Contention > Polling

   (All the average bandwidth request delays are very short.)

4. Average packet delays

   Polling >> Contention > Polling + Contention > Piggyback + Contention

Finally, we can know the most significant factor for average bandwidth requests per packet is the bandwidth request mechanisms, the most important factor for percentage of getting grant is the bandwidth request collision probabilities, and the most influential factor for average bandwidth request delays and average packet delays are the percentage of getting grant.

Now, we want to judge conclusions with the different bandwidth request mechanisms. In Heavy traffic load, we should use polling mechanism to request bandwidth because of the lower packet transmission time and the lower getting grant time (getting grant time means the average bandwidth request delay); in light traffic load, we should use contention mechanism and piggyback mechanism to request bandwidth. In addition, we prefer using piggyback mechanism than contention mechanism when we can use lots of piggyback requests.

At the section 4.2.5, we compare the different contention-based bandwidth request mechanisms, Type II - Contention mechanism and Type I - Contention mechanism. From the simulation results, we can know that in heavy traffic loads, Type II - Contention mechanism can reduce the total amount of bandwidth requests. However, in light traffic loads, Type II - Contention mechanism performs just like the Type I - Contention mechanism.

If we want to reduce the bandwidth requests sending by the SSs in heavy traffic loads, we could change Type I - Contention mechanism to Type II - Contention mechanism.

At the section 4.2.6, we compare the different piggyback-based bandwidth request mechanisms, M-Piggyback mechanism and piggyback mechanism. From the simulation results, we can know that in heavy traffic loads, M-Piggyback mechanism can reduce the average bandwidth request delays and average packet delays and will not affect the performance of rtPS and nrtPS connections. However, in light traffic loads, M-Piggyback mechanism performs just like the piggyback mechanism.

If we want to reduce the bandwidth request delays or the packet delays in heavy traffic loads, we could change piggyback mechanism to M-Piggyback mechanism.

Chapter 5. Conclusions and Future Work

In this thesis, we have discussed the different bandwidth request mechanisms on Best Effort connections, and we use four metrics (average bandwidth requests per packet, percentage of getting grant, average bandwidth request delays, and average packet delays) to these different bandwidth request mechanisms.

From the simulation results, we can know the most significant factor for average bandwidth requests per packet is the bandwidth request mechanisms, the most important factor for percentage of getting grant is the bandwidth request collision probabilities, and the most influential factor for average bandwidth request delays and average packet delays are the percentage of getting grant.

Finally, we can judge conclusions with the different bandwidth request mechanisms. In heavy traffic load, we should use polling mechanism to request bandwidth because of the lower packet transmission time and the lower getting grant time (getting grant time means the average bandwidth request delay); in light traffic load, we should use contention mechanism and piggyback mechanism to request bandwidth. In addition, we prefer using piggyback mechanism than contention mechanism when we can use lots of piggyback requests. If we want to use contention mechanism in heavy traffic load, we can use Type II - Contention mechanism to replace Type I - Contention mechanism by the reduction of the bandwidth requests, and if we want to use piggyback mechanism in heavy traffic load, we can use M-Piggyback mechanism to replace piggyback mechanism by the reduction of the bandwidth request delays and packet delays.

From our study, we know the system should use polling mechanism to get grant when the system is in the heavy traffic load, and should use contention/piggyback mechanisms to get grant when the system is in the light traffic load. These conclusions may help the IEEE 802.16 WiMAX MAC layer scheduler designers to design a more efficient scheduler for requesting bandwidth to increase the system performance.

In the future, we can study the side effects caused by the overheads of the grant management subheader in the M-Piggyback mechanism, and we also can study the impacts of bandwidth request mechanisms on the other types of scheduling services, like rtPS or nrtPS.

References

[1] IEEE Std. 802.16-2004, "Local and Metropolitan Area Networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems"

[2] Jianhua He, Ken Guild, Kun Yang, and Hsiao-Hwa Chen, "Modeling Contention Based Bandwidth Request Scheme for IEEE 802.16 Networks", *IEEE Communications Letters*, Volume 11, August 2007 pp.689-700.

[3] Vinel A., Ying Zhang, Qiang Ni, Lyakhov A., "Efficient Request Mechanism Usage in IEEE 802.16", *Global Telecommunications Conference*, December 2006, pp.1-5.

[4] Pries R., Staehle D., Marsico D.," Performance Evaluation of Piggyback Requests in IEEE 802.16", *Vehicular Technology Conference*, October 2007, pp.1892-1896.

[5] Qiang Ni, Vinel A., Yang Xiao, Turlikov A., Tao Jiang, "Investigation of Bandwidth Request Mechanisms under Point-to-Multipoint Mode of WiMAX Networks", *IEEE Communications Magazine*, Volume 45, Issue 5, May 2007, pp.132-138.

[6] Kameswari Chebrolu, Ramesh R. Rao, "Bandwidth aggregation for real-time applications in heterogeneous wireless networks", *IEEE Transactions on Mobile Computing*, April 2006, pp.388-403.

[7] Kameswari Chebrolu, Bhaskaran Raman, Ramesh R. Rao, "A Network Layer Approach to Enable TCP over Multiple Interfaces", *Wireless Networks*, Volume 11, Issue 5, September 2005, pp.637-650.

[8] Taleb T., Fernandez J.C., Hashimoto K., Nemoto Y., Kato N., "A Bandwidth Aggregation-aware QoS Negotiation Mechanism for Next-Generation Wireless Networks", *IEEE Global Telecommunications Conference*, November 2007,

pp.1912-1916.

[9] Fernandez J.C., Taleb T., Hashimoto K., Nemoto Y., Kato N., “Multi-path Scheduling Algorithm for Real-Time Video Applications in Next-Generation Wireless Networks”, *Innovations in Information Technology*, November 2007, pp.73-77.

[10] Liu Gan, Zhou Xin, Zhu Guang-xi, “A scheduling algorithm for maximum throughput based on the link condition in heterogeneous network”, *Journal of Communication and Computer*, March 2007, pp.33-37.

[11] Sung-Min Oh, Jae-Hyun Kim, “The Analysis of the Optimal Contention Period for Broadband Wireless Access Network”, *Pervasive Computing and Communications Workshops*, March 2005, pp.215-219.

[12] Cho D.-H., Song J.-H., Kim M.-S., Han, K.-J., “Performance analysis of the IEEE 802.16 Wireless Metropolitan Area Network”, *DFMA*, 2005, pp.130-137.

[13] Fei Yin, Guy Pujolle, “Performance Optimization for Delay-Tolerant and Contention-Based Application in IEEE 802.16 Networks”, *Journal on Wireless Communications and Networking*, 2008.

[14] Zsolt Saffer, Sergey Andreev, “Delay Analysis of IEEE 802.16 Wireless Metropolitan Area Network”, *Telecommunications*, June 2008, pp.1-5.

[15] Ben-Jye Chang, Chien-Ming Chou, “Markov Chain Model for Polling Delay and Throughput Analyses of Uplink Subframe in WiMAX Networks”, *Wireless Communications and Networking Conference*, April 2008, pp.1368-1373.