國立交通大學

網路工程研究所

碩士論文

利用封包聚合機制增進 IEEE 802.11 無線電感知網路之效能

Using Frame Aggregation Mechanism to Improve the Performance of IEEE 802.11 Cognitive Radio Network

研 究 生：王柏凡

指導教授：王協源 教授

中華民國一百零一年十月

利用封包聚合機制增進 IEEE 802.11 無線電感知網路之效能

Using Frame Aggregation Mechanism to Improve the Performance of IEEE 802.11 Cognitive Radio Network

研究生：王柏凡 Student：Po-Fan Wang

指導教授：王協源 Advisor：Shie-Yuan Wang

國立交通大學

網路工程研究所

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

Computer Science

October 2012

Hsinchu, Taiwan, Republic of China

中華民國一百零一年十月

利用封包聚合機制增進 IEEE 802.11 無線電感知網路之效能

學生：王柏凡　　　　　　　　　　指導教授：王協源　教授

國立交通大學 網路工程研究所 碩士班

摘要

隨著科技進步以及個人通訊裝置的興起，近年來無線網路的應用在種類以及數量上蓬勃發展，現有的頻帶多採用固定劃分的方式，由業者向政府機構申請職執照後發配，有研究指出此種固定劃分的機制造成整體的頻譜使用效率不佳，原因是部分所劃分出去的頻帶使用率不高，然而其他無執照的使用者並不能使用該頻帶。整體而言，可用於傳輸之無線網路頻譜逐漸感到不足，因此，近年來無線電感知網路的領域蓬勃發展，此技術的提出宗旨在於更有效率的利用無線頻譜，其中有一部份的機制可以達到在不干擾已劃分之頻段的主要使用者的情況下進而使用該頻段。

本篇論文主要繼承前人在 IEEE 802.11 無線電感知網路的研究成果，進行在更高資料傳輸率以及更新的主要使用者情況下進行驗證，並利用 IEEE 802.11n 封包聚合機制來改善現有機制的效能並且透過更新版的 Estinet 網路模擬器來驗證效能的提升以及對主要使用者的影響。

關鍵字：無線感知網路、網路模擬器、IEEE 802.11n 封包聚合機制

Using Frame Aggregation Mechanism to Improve the Performance of IEEE 802.11 Cognitive Radio Network

Student: Po-Fan Wang Advisor: Shie-Yuan Wang

Department of Network Engineering

National Chiao Tung University

ABSTRACT

The concept of cognitive radio (CR) has become more and more popular and important due to the limitation of wireless bandwidth. The current wireless spectrum policy separates the wireless spectrum into many separated spectra and defines them into licensed band or non-licensed band. However, a research shows that the usage of licensed spectrum is less than 25%, which is very inefficient. In a CR network, the CR users are allowed to use the empty spectra in frequency, time and space under the constraints of not interfering with the primary users. The CR approach has a great potential to improve the utilization of licensed spectrum. In this thesis, we follow the previous research of an 802.11 Cognitive Radio Network and use the IEEE 802.11n frame aggregation mechanism to improve the performance and evaluate the result on Estinet network simulator. In our simulation result, we show that the frame aggregation mechanism successfully improves the performance of previous work and the frame aggregation mechanism does not increase the influence to PUs.

Keyword: Cognitive Radio, Network Simulator, IEEE 802.11n Frame Aggregation Mechanism

誌　謝

首先感謝我的指導教授王協源老師這兩年來的細心指導，以及給我在學術上的訓練，並讓我從一進研究所就開始有機會能夠將研究成果發表至國際期刊，這些都是我在人生中重要的成就。此外，感謝三位口試委員李端興老師、吳曉光老師、趙禧綠老師擔任我的口試委員並給予許多很好的建議以及指導，使我的論文更完整、豐富。

感謝所有 NSL 實驗室的成員，感謝上一屆的學長們在畢業最後關頭給了我畢業的勇氣與信心，感謝同屆同學豐富了我兩年的實驗室生活，與你們健身聊天將是我一輩子的回憶，感謝強者學弟妹帶來認真的風氣，讓實驗室的氣氛止跌回升，感謝碩二下時來自對岸的交換生，讓我在學生生涯的最後階段能夠有精采的回憶。

最重要的是感謝我的父母，從小到大當我堅實的後盾以及溫暖的避風港，並在研究所的最後階段毫無保留的支持我，讓我無後顧之憂地向前衝刺。

感謝待了六年的交大以及新竹，我畢業了！

王柏凡　2012.10 秋

目　　　錄

中　文　摘　要 I

英　文　摘　要 III

目　　　錄 V

圖　目　錄 VII

表　目　錄 XI

Chapter 1　Introduction 1

Chapter 2　Background Introduction to Cognitive Radio Network 3

- 2.1.　Introduction to Cognitive Radio Network 3
- 2.2.　Throughput Limitation of 802.11 WLAN 4
- 2.3.　IEEE 802.11n Frame Aggregation and Block ACK Mechanisms 6
  - 2.3.1.　Aggregated-Mac Service Data Unit 7
  - 2.3.2.　Aggregated-Mac Protocol Data Unit 8
  - 2.3.3.　Two-Level Aggregation (TLA) 10
  - 2.3.4.　Block ACK Mechanism 11

Chapter 3　Related work 13

- 3.1.　Related work I: On Synchronized Channel Sensing and Accessing for Cognitive Radio Users in IEEE 802.11 Wireless Networks (Ori-MAC) 13
- 3.2.　Related work II: Enhanced MAC Protocol for Cognitive Radios over IEEE 802.11 Networks (Uni-MAC) 19
- 3.3.　Related work III: Bi-directional Cognitive Radio MAC Protocol for Supporting TCP Flows (Bi-MAC) 24

3.4. Related work IV: An IEEE 802.11 cognitive radio MAC protocol with dynamic bandwidth allocation capabilities. 30
Chapter 4 Motivation 40
Chapter 5 Proposed Frame-aggregated CR Mac protocol (FA-MAC) 42
5.1. Migration of Uni-MAC and ABi-MAC over Estinet Network Simulator 43
5.2. New Dynamic Bandwidth Allocation Process (NDBA) 44
5.3. New Bandwidth Negotiation Process 47
5.4. Implementation of Aggregated-Mac Service Data Unit (A-MSDU) Mechanism 50
5.5. Implementation of Aggregated-Mac Protocol Data Unit (A-MPDU) Mechanism 52
5.6. Implementation of Block ACK Operation 57
Chapter 6 Performance evaluation 58
6.1. Simulation Settings 59
6.2. The Improvement of Maximum Throughput of FA-MAC 62
1. 6.3. . The Effect on Primary Users' Throughput and Packet Transmission Delay from FA-MAC 70
6.4. Summary 83
Chapter 7 Conclusion 84
Chapter 8 REFERENCES 86

圖 目 錄

Figure 2.1 Normalized overhead under different data rate and payload in [5] 5

Figure 2.2 The Maximum Throughput and the Throughput Upper Limit for IEEE 802.11a in [5] 6

Figure 2.3 Aggregated-Mac Service Data Unit 8

Figure 2.4 Aggregated-Mac Protocol Data Unit........................10

Figure 2.5 Two-Level Aggregation 11

Figure 2.6 Illustration of Block ACK operation........................12

Figure 3.1 Illustration of Uni-MAC protocol with $TXOP_{CR}$ value is 2........................18

Figure 3.2 Illustration of Hidden Terminal Problem 19

Figure 3.3 Illustration of AR update procedure and Calculation of the Availability Index.....20

Figure 3.4 Frame format of RTS_{CR} in Uni-AMC 21

Figure 3.5 Frame format of CTS_{CR} in Uni-AMC 21

Figure 3.6 Illustration of Uni-MAC and the enhancement when $TXOP_{CR}$ value is 2............24

Figure 3.7 Illustration of the best case of the TCP packets exchanges when $TXOP_{CR}$ is 2 26

Figure 3.8 Illustration of a bad case of the TCP packets exchanges when $TXOP_{CR}$ is 2........27

Figure 3.9 Illustration of Packets Exchanges on Data Channel of a) Uni-MAC and b) Bi-MAC, when $TXOP_{CR}$ is 2........................30

Figure 3.10 Illustration of Message Exchanges on Data Channel of Bi-MAC with Bi-directional Bandwidth Reservation when $TXOP_{CR}$ = 3 33

Figure 3.11 Illustration of inappropriate TI setting of Bi-MAC with Bi-directional Bandwidth Reservation........................35

Figure 3.12 Frame Format of the control packet RTS_{CR} of ABi-MAC 37

Figure 3.13 Frame Format of the control packet CTS_{CR} of ABi-MAC..................................37

Figure 5.1 Illustration of IEEE 802.11 DCF implementation on NCTUns and Estinet. 44

Figure 5.2 Illustration of NDBA without switching the sender/receiver ID when $TXOP_{CR}$ is 6 46

Figure 5.3 Switching Sender/Receiver ID Case of NDBA when $TXOP_{CR}$ is 4 47

Figure 5.4 Illustration of Bandwidth Negotiation Process 49

Figure 5.5 Illustration of New Bandwidth Negotiation Process...50

Figure 5.6 Illustration of CRU Protocol Modules Stack with A-MSDU module. 52

Figure 5.7 The Structure of frame_item And frame_pack. 54

Figure 5.8 Illustration of A-MPDU frame aggregation and NDBA with asymetric flow. 55

Figure 5.9 Illustration of A-MPDU And Block ACK Operation Over Uni-MAC ($TXOP_{CR}$ = n) 56

Figure 5.10 Illustration of A-MPDU and Block ACK operation over ABi-MAC with symmetric traffic flow ($TXOP_{CR}$ = 6) 56

Figure 5.11 The Structure of frame_item and frame_pack..57

Figure 6.1 Simulation Case 1 for Maximum Throughput Evaluation....................................60

Figure 6.2 Simulation case 2 for evaluating the influence to PUs 61

Figure 6.3 Maximum UDP Throughput of Uni-MAC with different Frame Aggregation Mechanism (with UDP payload 1450 bytes)..64

Figure 6.4 Maximum TCP Throughput of Uni-MAC with different Frame Aggregation Mechanism 65

Figure 6.5 Maximum UDP Throughput of ABi-MAC with different Frame Aggregation Mechanism (with UDP payload 1450 bytes)..66

Figure 6.6 Maximum TCP Throughput of ABi-MAC with different Frame Aggregation Mechanism 68

Figure 6.7 Maximum Application Layer UDP Throughput of ABi -MAC with different Bit Error Rates (with UDP payload 1450 bytes, TXOP_{CR} = 10) 69

Figure 6.8 Maximum Application Layer TCP Throughput of ABi-MAC with different Bit Error Rates (TXOP_{CR} = 10) 70

Figure 6.9 UDP Throughput of CRU with 30% PU Load 73

Figure 6.10 PU’s Average Packet Delay Time (30% PU Load) 73

Figure 6.11 TCP Throughput of CRU with 30% PU Load 74

Figure 6.12 PU’s Average Packet Delay Time (30% PU Load) 74

Figure 6.13 Application Layer UDP Throughput of ABi-MAC with Different Bit Error Rates (With UDP payload 1450 bytes, TXOP_{CR} = 10) 75

Figure 6.14 Application Layer TCP Throughput of ABi-MAC with different Bit Error Rates (TXOP_{CR} = 10) 75

Figure 6.15 UDP Throughput of CRU with 50% PU Load 77

Figure 6.16 PU’s Average Packet Delay Time (50% Load) 77

Figure 6.17 TCP Throughput of CRU with 50% PU Load 78

Figure 6.18 PU’s Average Packet Delay Time (50% Load) 78

Figure 6.19 Application Layer UDP Throughput of ABi-MAC with different Bit Error Rates (UDP payload 1450 bytes, TXOP_{CR} = 10, 50% PU Load) 79

Figure 6.20 Application Layer TCP Throughput of ABi-MAC with different Bit Error Rates (TXOP_{CR} = 10, 50% PU Load) 79

Figure 6.21 The Throughput of PUs with the effect of CRUs (80% PU Load) 81

Figure 6.22 UDP Throughput of CRU with 80% PU Load 81

Figure 6.23 PU’s Average Packet Delay Time (80% PU Load) 81

Figure 6.24 The Throughput of PUs with the effect of CRUs (80% PU Load) 82

Figure 6.25 TCP Throughput of CRU with 80% PU Load 82

Figure 6.26 PU’s Average Packet Delay Time (80% PU Load) 82

表　目　錄

Table 6.1 The abbreviation of different CR MACs and Frame aggregation Mechanisms 58

Table 6.2 The Parameter Setting of CR MAC protocol 59

Table 6.3 Traffic Flows Setting of Case 1 60

Table 6.4 PUs Traffic Flow and the configuration command......61

Table 6.5 The Network Flow applied to CRUs 62

Table 6.6 The Improvement Of Maximum UDP Throughput When Different Frame Aggregation Mechanism Is Applied. (in percentage)......64

Table 6.7 The Improvement Of Maximum TCP Throughput When Different Frame Aggregation Mechanism Is Applied (in percentage)......65

Table 6.8 The Improvement of Maximum UDP Throughput When Different Frame Aggregation Mechanism Is Applied (in percentage)......66

Table 6.9 The Improvement Of Maximum TCP Throughput When Different Frame Aggregation Mechanism Is Applied. (in percentage)......68

Table 6.10 The PU’s Throughput of 30% Load with the effect of CRUs (KB/s)......73

Table 6.11 The PU’s Throughput of 30% Load with the effect of CRUs (KB/s)......74

Table 6.12 The PU’s Throughput of 50% Load with the effect of CRUs (KB/s)......77

Table 6.13 The PU’s Throughput of 50% Load with the effect of CRUs (KB/s)......78

Chapter 1 Introduction

The concept of cognitive radio (CR) has become more and more popular and important due to the limitation of wireless bandwidth. In [13], the authors show that the usage of licensed spectrum is less than 25%, which is very inefficient. In a CR network, the secondary/unlicensed users (CRUs) are allowed to use the empty spectra in frequency, time and space under the constraints of not interfering with the primary/licensed users (PUs). The CR approach has a great potential to improve the utilization of licensed spectrum. For licensed band Cognitive Radio, there are two important standards: IEEE 802.22 [9] and ECMA-392 [10]. In contrast, IEEE 802.15.2 working group [12] which is working on the coexistence mechanisms of IEEE 802.15 and IEEE 802.11 is one of the unlicensed-band Cognitive Radio.

This thesis is an extension of the work done by [2], [3] and [4]. Shie-Yuan Wang et al. [2] proposed a CR MAC protocol that allows CRUs to access the idle period of an IEEE 802.11 network with a *Smart Channel Sensing Scheme* and the protection from hidden terminal. Through extensive simulations, Shie-Yuan Wang et al. found that the Transmission Control Protocol (TCP) did not perform well under their extended work. Lee-Chin Lau et al. in [3] and [4] proposed a Bi-directional CR MAC protocol to solve the TCP problem. However, they use old fashion IEEE 802.11b stations as PUs and the data rate applied in the simulation is 2 Mbps. We think those work should be examined and verified under the conditions of using higher data rate and using a more advanced version of IEEE 802.11 stations as PUs.

Furthermore, we consider an improvement to enhance the efficiency of CR-MAC protocol through frame aggregation mechanism which is proposed in IEEE 802.11n [8]. The frame aggregation mechanism is already proved that it does improve the efficiency of IEEE 802.11 [6]. We can expect that frame aggregation mechanism could also enhance the efficiency of CR-MAC protocol. In this paper, we implement the frame aggregation mechanism on both Uni-MAC [2] and ABi-MAC [4] to increase the available throughput and analyze the effects of the enhancement on PUs.

The remainder of this dissertation is organized as follows. We will present some background knowledge in Chapter 2. In Chapter 3, the major related work is elaborated. The motivation is mentioned in Chapter 4 and the design and implementation of frame aggregation mechanism on Uni-MAC and ABi-MAC is explained in Chapter 5. Later in Chapter 6, we use the Estinet network simulator to verify our design and see the effect of frame aggregation mechanism on Uni-MAC and ABi-MAC. Finally, we will present our conclusion in Chapter 7.

Chapter 2 Background Introduction to Cognitive Radio Network

2.1. Introduction to Cognitive Radio Network

Recently, the concept of cognitive radio (CR), which was first introduced by Joseph Mitola III [9], has become more and more popular and important due to the limitation of wireless bandwidth. The current wireless spectrum policy separates the wireless spectrum into many separated spectra and defines them into licensed band or non-licensed band. One can tell the meaning by name that licensed band is only allowed for licensed users in order to guarantee the application and communication quality on it. On the other hand, non-licensed band is free for everyone but the user on it has to contend with each other for the bandwidth. An example for licensed band is Wireless TV band (54 MHz and 806 MHz in U.S.) and an example for non-licensed band is 2.450 GHz for WLAN.

Unlike the current policy, cognitive radio has the ability to detect available channel in wireless medium, switch itself to the channel, and changes the transmission and reception parameter automatically so that more wireless communications may run concurrently in a given spectrum band at a place.

For licensed band Cognitive Radio, there are two important standards: IEEE 802.22 [10] and ECMA-392 [11]. Both of them defined the MAC and PHY specification of Cognitive Radio network that operate on TV White Space (this unused TV spectrum is often termed as "white spaces") of Wireless TV band. In contrast with licensed band Cognitive Radio, there is also Unlicensed-Band Cognitive Radio which can only utilize unlicensed parts of the radio frequency (RF) spectrum. An example of unlicensed-band Cognitive Radio is IEEE 802.15.2™-2003 [12], which defines on the coexistence mechanisms of IEEE 802.15 and IEEE 802.11.

2.2. Throughput Limitation of 802.11 WLAN

In the last decade, wireless network technology has changed our daily life significantly. One of the most popular wireless networks is IEEE 802.11 wireless local area network (WLAN). Its success is due to some good characteristics such as mobility, flexibility, and low cost. Although the original IEEE 802.11 specification provides only 2 Mb/s date rate, the later IEEE 802.11b, and 802.11a/g specifications provide up to 11 Mb/s and 54 Mb/s data rates. However, the improvement of physical layer data rates does not increase the user-level throughput due to the design of IEEE 802.11 MAC protocol. In [5], the authors analyze the overhead of IEEE 802.11 MAC under different data rate at different payload sizes. They indicate that the overhead is the major issue of inefficient MAC includeing headers (MAC header, frame check sequence [FCS], and PHY header), interframe spaces (IFSs), backoff time, and ACKs. They defined overhead as the difference between data rate and throughput and defined normalized overhead as overhead divided by data rate. Figure 2.1 shows that the normalized overhead reaches over 90% at 50 Mbps data rate when the payload size is 100 bytes and the normalized overhead still reaches over 40% at 50 Mbps data rate when the payload size is 1,500 bytes.

Figure 2.2 shows the maximum throughput and throughput upper limit of IEEE 802.11a. The throughput upper limit (TUL) is defined as the maximum throughput when the raw data rate goes infinitely high. In Figure 2.2, the maximum throughput reaches about 30 Mbps with the 54 Mbps data rate and 1,500 bytes payload. However, the maximum throughput is about only 54 Mbps with the 216 Mbps data rate and 1,500 bytes payload. In this case, the used data rate is four times higher but the throughput improvement is less than twice. In order to overcome the overhead issue of IEEE 802.11 MAC, IEEE 802.11n is proposed with several efficient MAC enhancements, which that we will address in the following sections.

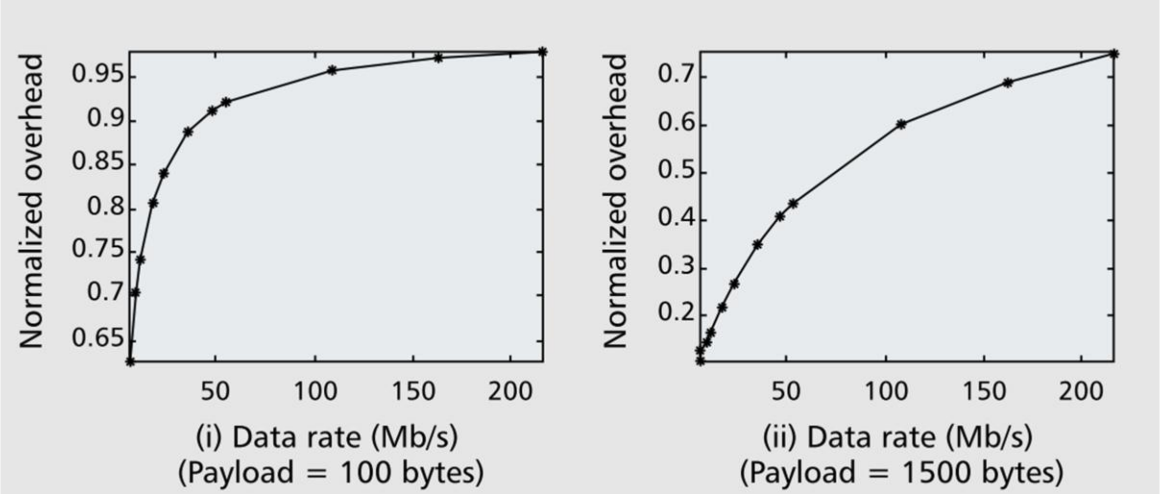


Figure 2.1 Normalized overhead under different data rate and payload in [5]   Y. Xiao,

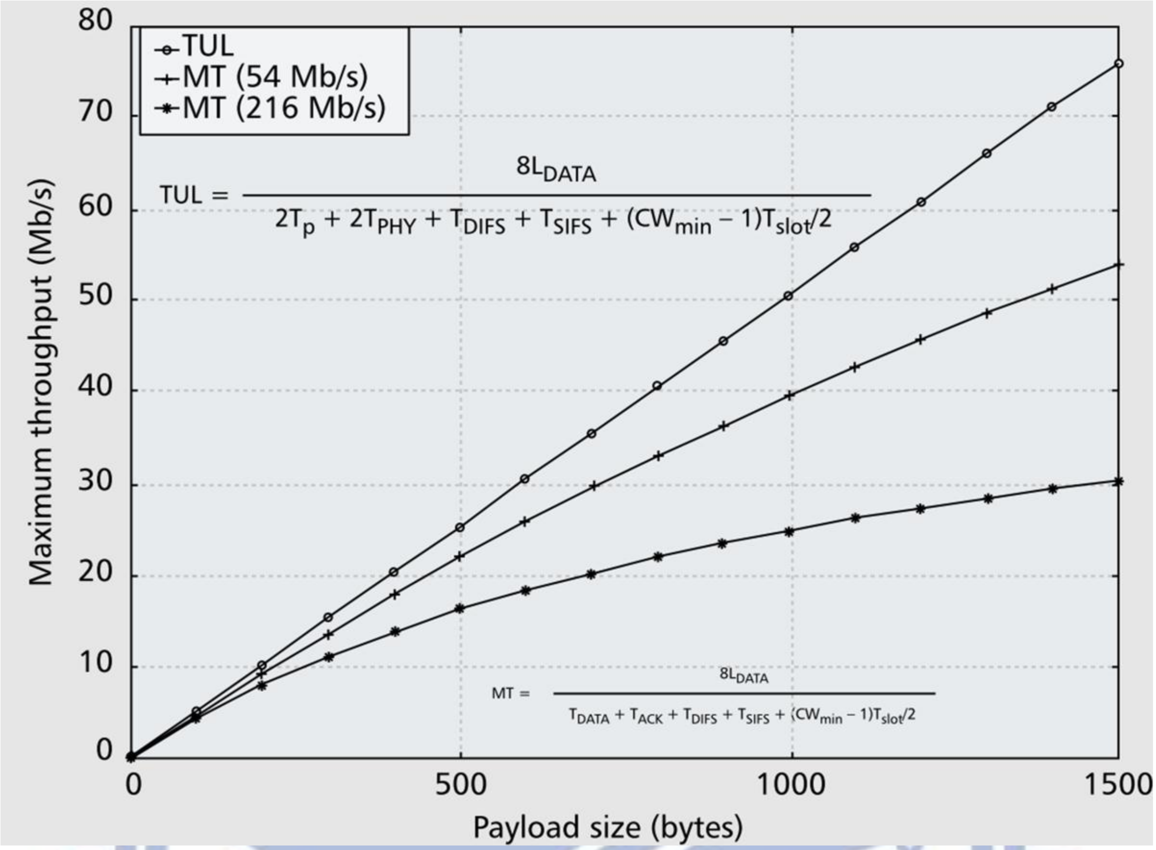


Figure 2.2 The Maximum Throughput and the Throughput Upper Limit for IEEE 802.11a in [5] Y. Xiao, “IEEE 802.11n: Enhancements for Higher Throughput in Wireless

2.3. IEEE 802.11n Frame Aggregation and Block ACK Mechanisms

Before the introduction of frame aggregation mechanisms, there are some terms that readers should know. The IEEE Std 802.11™-2007 [7] defines:

- *Medium access control (MAC) service data unit (MSDU)*: information that is delivered as a unit between MAC service access points (SAPs).
- *Medium access control (MAC) protocol data unit (MPDU)*: the unit of data exchanged between two peer MAC entities using the services of the physical layer (PHY).

- *PLCP protocol data unit (PPDU)*: the actual frame send on the wireless medium.

2.3.1. Aggregated-Mac Service Data Unit

Figure 2.3 illustrates the concept of A-MSDU aggregation. As one can see, this method aggregates the MSDU sending from the upper layer of MAC protocol layer into one big A-MSDU frame. Only those MSDUs that has the same destination address (DA) and source address (SA) as the receiver address (RA) and transmitter address (TA) can be aggregated into one A-MSDU. To completely form an A-MSDU either when the size of the waiting packets reaches the maximal A-MSDU threshold or the maximal delay of the oldest packet reaches a pre-assigned value. After the aggregation, the MSDUs become sub frames of the A-MSDU and send in to mac protocol layer. In the mac protocol layer, an A-MSDU should be encapsulated into a single MPDU. All the A-MSDU sub frames will share the same MAC header after being aggregated into one MPDU. One important point is that an A-MSDU in a MPDU should not be fragmented and the destination address field in mac header of the MPDU carrying an A-MSDU should be set to an individual address. One should know there is a size limitation of A-MSDU. In IEEE 802.11n [8], the maximum size of A-MSDU is ether 3389 bytes or 7935 bytes. A mobile station should not transmit an A-MSDU to another mobile station that exceeds its maximum A-MSDU length capability.

The A-MSDU aggregation technique reduces the overhead of the mac header by using a mac header to all the A-MSDU sub frames. A bigger improvement to the IEEE 802.11 mac protocol is it reduces the transmission overhead by sending the aggregated MSDU together rather than sending them one at a time.

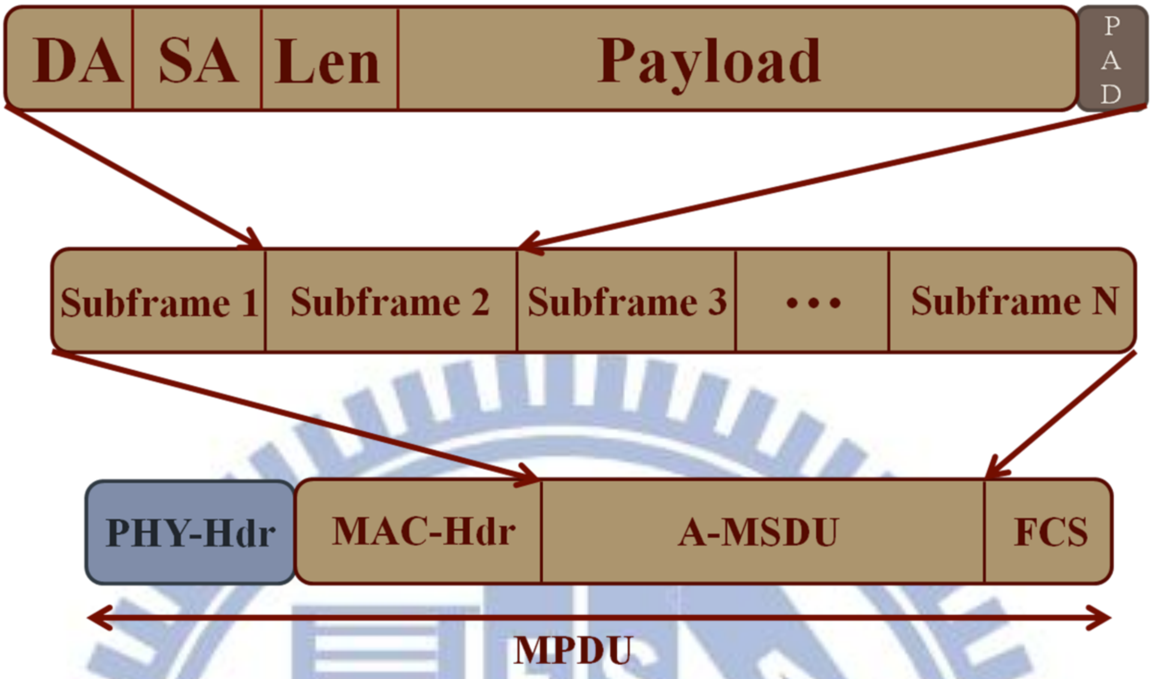


Figure 2.3 Aggregated-Mac Service Data Unit

2.3.2. Aggregated-Mac Protocol Data Unit

Figure 2.4 illustrates the concept of A-MPDU aggregation. The idea of A-MPDU aggregation is to join multiple MPDU sub frames with a single leading PHY header. The A-MPDU aggregation is done after the MAC header encapsulation process. The A-MPDU aggregation restriction factor doesn’t contain the TID of the MPDU frame which is different from A-MSDU aggregation. Still, all the aggregated MPDUs must be addressed to the same receiver address. There is no waiting time to form an A-MPDU which means an A-MPDU is formed by the packets already in the transmission queue at the aggregation period. The maximum length of an A-MPDU is 65,535 bytes and the maximum number of sub frames that it can hold is 64 because a block ACK bitmap field is 128 bytes in length, where each frame is mapped using two bytes. The two bytes are for the acknowledgement of up to 16 fragments but because A-MPDU does not allow fragmentation, these extra bits are excessive. There is a new variant so called compressed block ACK which has a bitmap field only eight bytes long. One can see there is a set of fields, known as MPDU delimiters in Figure 2.4. The delimiters are inserted before each MPDU and a set of padding bits length from 0 – 3 bytes are added at the tail. The delimiter header is used to define the MPDU position and length inside the aggregated frame. The cyclic redundancy check (CRC) field in the delimiter verifies the authenticity of the 16 preceding bits. The padding bytes are added such that each MPDU is a multiple of four bytes in length, which can assist sub frame delineation at the receiver side. In other words, the MPDU delimiters and PAD bytes determine the structure of the A-MPDU. After the A-MPDU is received, a de-aggregation process initiates. First it checks the MPDU delimiter for any errors based on the CRC value. If it is correct, the MPDU is extracted, and it continues with the next sub frame until it reaches the end of the PSDU. Otherwise, it checks every four bytes until it locates a valid delimiter or the end of the PSDU. The delimiter signature has a unique pattern to assist the de-aggregation process while scanning for delimiters.

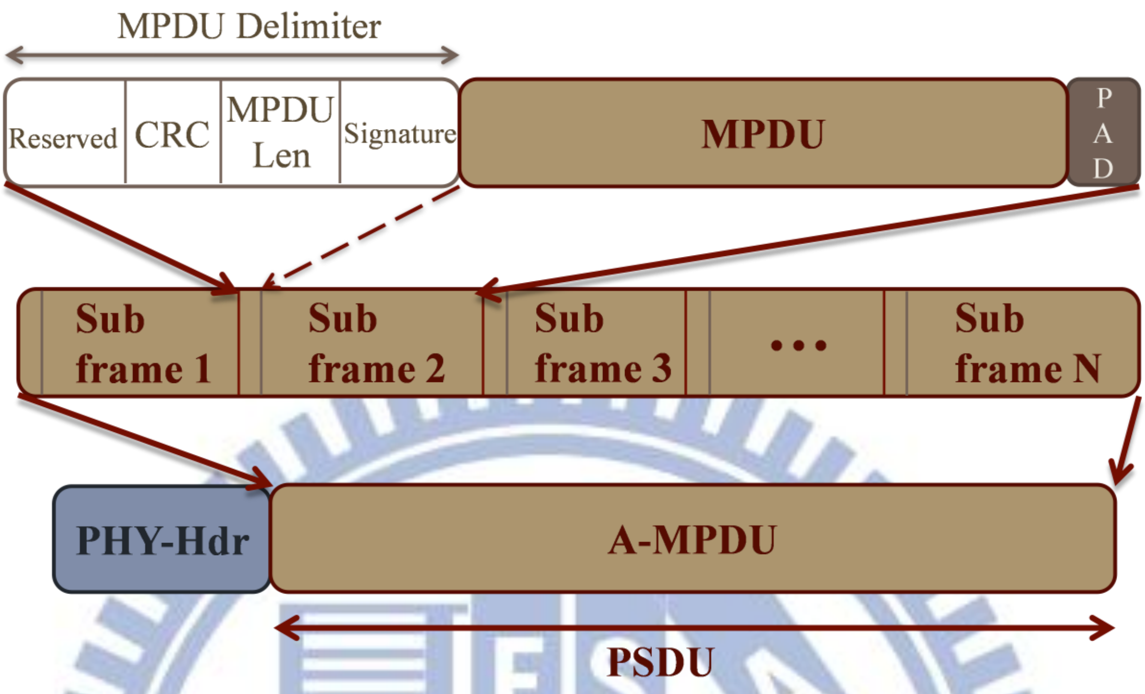


Figure 2.4 Aggregated-Mac Protocol Data Unit

2.3.3. Two-Level Aggregation (TLA)

Using A-MSDU and A-MPDU together called two-level frame aggregation. Figure 2.5 illustrates the concept of two-level frame aggregation. In first level, when a MSDU coming down from the upper layer, it first buffered in the A-MSDU provisional storage area and check if it satisfy the A-MSDU aggregation condition explained in the previous related subsection. If a data unit satisfied the aggregation condition, it can be compacted in to an A-MSDU. If condition is not satisfy such as TID is different, all these aberrant frames can move to the next level where they will be packed together with any A-MSDUs derived from the first level or other single MSDUs by using A-MPDU aggregation.

A two-level frame aggregation comprises a blend of A-MSDU and A-MPDU over two stages. In Figure 2.5 we illustrate how this new scheme can be achieved. The basic operation is explained as follows: In the first stage, if any MSDUs that are buffered in the A-MSDU provisional storage area justify the A-MSDU constraints explained in the previous related subsection, these data units can be compacted into a single A-MSDU. If the TIDs are different, all these aberrant frames can move to the second stage where they will be packed together with any A-MSDUs derived from the first stage or other single MSDUs by using A-MPDU aggregation. However, it must be mentioned that given that the maximum MPDU length for an A-MPDU data frame is limited to 4095 bytes, then A-MSDUs or MSDUs with lengths larger than this threshold cannot be transmitted. Conjointly, any fragments from an A-MSDU or MSDUs also cannot be included in an A-MPDU. In the following section, we evaluate how this synthesis is more efficient in most of the cases than A-MPDU and A-MSDU aggregation operating alone.

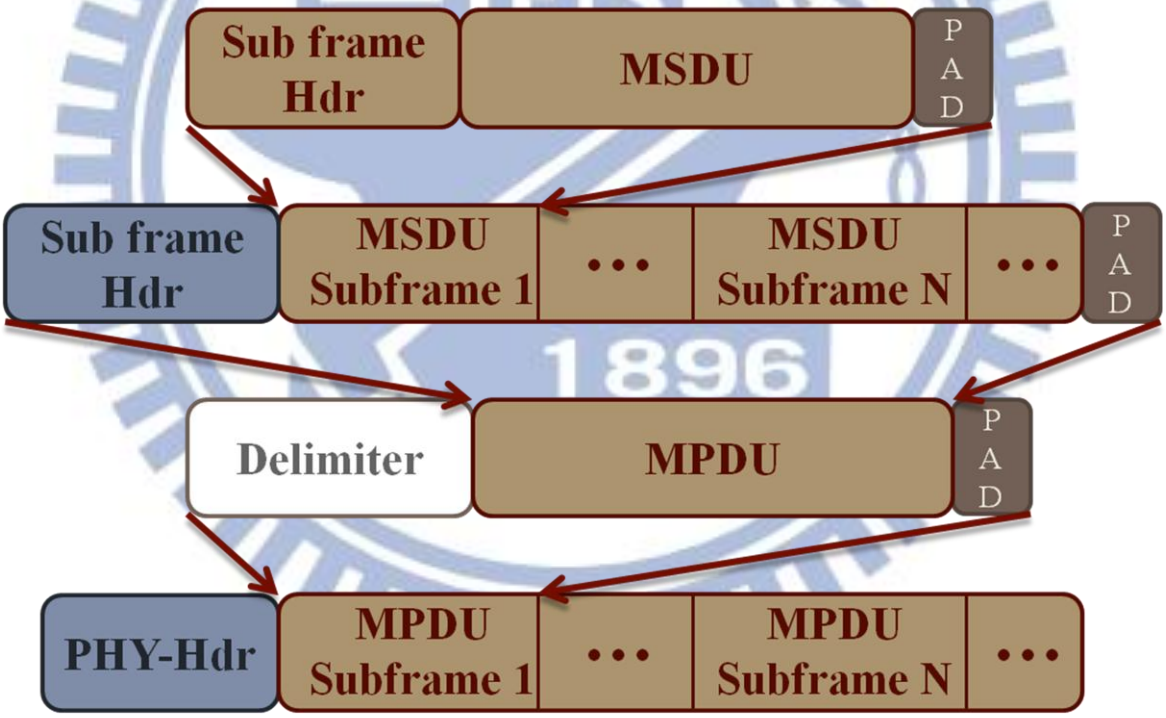


Figure 2.5 Two-Level Aggregation

2.3.4. Block ACK Mechanism

The Block ACK mechanism is proposed in IEEE 802.11e in [7]. It improves the channel efficiency by saving the time wasted from ACK transmission. The station can transmit several

PPDU separated by SIFS and acknowledged by a single aggregated ACK frame, i.e. the *Block ACK* (BA) frame. The BA acknowledges the received frame with a bitmap field which contains the information about the reception of corresponding MPDUs. The BA is reply after receives a block ACK request (BAR) message. Both the block ACK request and block ACK frame are transmitted at the same transmission rate that is used for the corresponding data frame transmission. The A-MPDU aggregation should work with Block ACK mechanism in order to transmit the PPDU frames burst.

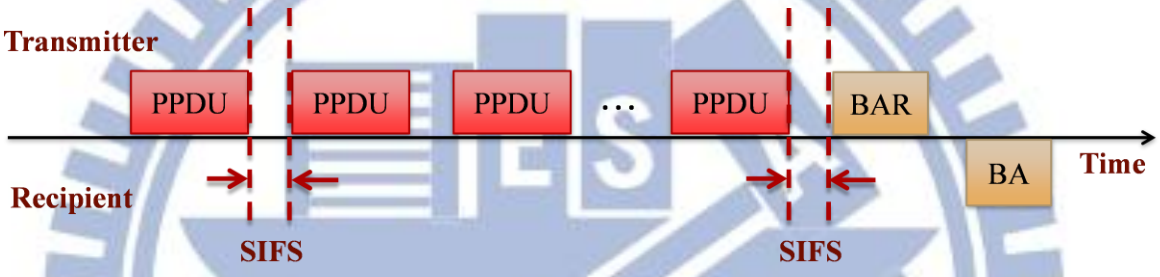


Figure 2.6 Illustration of Block ACK operation.

Chapter 3 Related work

In this chapter, four major related work is introduced in the following section: Related Work I, Related Work II, Related Work III, and Related Work IV. For the simplicity of the article, we call the CR MAC protocol of Related Work I: Ori-MAC, the CR MAC protocol of Related Work II: Uni-MAC, the CR MAC protocol of Related Work III: Bi-MAC and the CR MAC protocol of Related Work IV: ABi-MAC in the following.

3.1. Related work I: On Synchronized Channel Sensing and Accessing for Cognitive Radio Users in IEEE 802.11 Wireless Networks (Ori-MAC)

This paper proposed a synchronized channel-sensing and accessing mechanism for cognitive radio users in IEEE 802.11 wireless networks. The mechanism consists of two phases: *fast channel sensing* and *proactive channel vacating*. Fast channel sensing mechanism is for a pair of cognitive radio users to find an available channel efficiently; proactive channel vacating mechanism provides the protection of the channel usage of primary users.

- New control frames and system parameters:

  The authors define three control frames and four parameters:

  New control frames:

  - **$\mathrm{RTS}_{\mathrm{CR}}$:** this control frame is used while a CRU (denoted as CR sender) wants to initiate a data transmission process with another CRU (denoted as CR receiver). The control frame carry two channel hopping information: initial selected data channel ID and increment-per-hop. The two parameters are introduced in the new parameters section below. RTS$_{CR}$ control frame is sent on control channel only.

- **CTS$_{CR}$:** this control frame is used for a CR receiver to replay its CR sender that it is idle and ready to start the channel sensing procedure. CTS$_{CR}$ control frame is also sent on control channel only.
- **Ready-To-be-Interrupted (RTI):** this control frame is used while CR sender receives an ACK for a CR receiver after a successful packet transmission. CR sender sends a RTI for primary users nearby to claim the data channel if they is data available to send.

New parameters:

- **Initial selected data channel ID (denoted as Ch(1)):** it is the first data channel that selected by CR sender randomly. The CR pair will first switch to this channel to do the channel sensing. *Ch(1)* is a positive integer, and $1 \leq Ch(1) \leq N$.
- **Increment-per-hop (*h*):** is the hopping parameter for CRUs to switch to other channel. When CR sender and receiver sense PUs signal during the sensing period on *Ch(1)*. The CRUs will switch to the next channel *Ch(1) + h*. *h* is smaller than data channel number (*N*) and *h* and *N* are relative primes.
- **TXOP$_{CR}$:** is the number of frames that a CR sender can send on a data channel.
- **SIFS$_{CR}$:** is a small time interval between RTI and the next data frame. The function of SIFS$_{CR}$ is providing opportunities for PUs to claim the channel. When one PU with data frames to send, it sends out RTS after receiving RTI. CRUs will detect the presence of PUs and return to control channel. The SIFS$_{CR}$ should longer than time for a PU to sends out the RTS which is set to 10 * SIFS.

- Fast channel sensing:

When a CR sender want to send some data frames to a CR receiver, CR sender should first transmits a RTS$_{CR}$ control frame which carrying the channel hopping information (initial selected data channel ID and increment-per-hop) to CR receiver. After receiving RTS$_{CR}$ control frame, CR receiver reply a CTS$_{CR}$ frame to inform CR sender that it is in idle state and ready to receive data. CR receiver then switch to initial selected data channel *Ch(1)*. CR sender also switch to *Ch(1)* after receiving the CR receiver's reply. We call the exchange of RTS$_{CR}$ and CTS$_{CR}$ as **Bandwidth Negotiation Process** (BNP) in the following sections.

After switching into *Ch(1)* data channel, Both CR sender and CR receiver start to listen to the channel for 2ms. During this sensing time interval, the CR pair listens to the channel to avoid the interference of PUs' data transmission. If the data channel is heard idle, CR sender and CR receiver will start the data transmission process. Otherwise, CR sender and CR receiver hop to the next channel and start to sense again. The data transmission process starts with an RTS/CTS exchange between CR sender and CR receiver. This avoids a collision when more than one pair of CRUs switches into same data channel at the same time. The next channel ID is decided by (3-1).

$$\mathrm{Ch}(\mathrm{i}+1) = (\mathrm{Ch}(\mathrm{i}) + \mathrm{h}) \bmod \mathrm{N}, \mathrm{i} \geq 1, \qquad (3\text{-}1)$$

where *Ch(1)* is initial selected data channel ID and i mean the ith channel hopping. This function can generate a hopping sequence that going through all data channels without entering any channel repeatedly. For instance, the number of data channel N is 8 (numbered 0-7); *Ch(1)* and *h* are 4 and 3, the complete hopping sequence will be [4, 7, 2, 5, 0, 3, 6, 1, 4, 7, 2, …].

After the BNP, a CR pair will try to access each data channel in sequence generated by (3-1) until finding an idle channel and do the data transmission. Otherwise, the CR pair will keep trying the next channel until the CR pair finds one data channel which is not occupied by PUs and other CRUs.

If a CR pair successfully finds an idle data channel after channel sensing, CR sender sends a RTS frame to CR receiver instantly and waits for CR receiver’s reply. Once the CR sender receives the reply successfully, CR sender starts to transmit the data frames one at a time. After receiving the ACK reply from CR receiver, CR sender sends the broadcast RTI frame to one-hop neighboring PUs and wait for $\mathrm{SIFS}_{\mathrm{CR}}$ time so that PUs has the opportunity to claim the channel. We called this time interval as **Quiet Period** (QP). If CR sender receives a RTS from PU, the CR pair then vacates the data channel and hop back to control channel. If no RTS is received, CR sender can transmit another data frame to CR receiver after QP. If no PU claims the data channel after each RTI sent by CR sender, CR sender can transmit upto $\mathrm{TXOP}_{\mathrm{CR}}$ frames. The RTI control message will inform that if there is more date frame to CR receiver after RTI and QP. After that, the CR pair need to hop back to control channel even there are more data frames in the queue. If there are still queued frames to send, the CR pair must run the procedure of BNP and fast channel sensing again. We defined the process (a CRU pair start BNP until they hop back to control channel) above as “**CR Transmission Round”** in the following of this paper.

➢ Proactive channel vacating:

There are two kind of situations mention above for a CRU pair to vacate the data channel.

- The CRU pair already transmitted $\mathrm{TXOP}_{\mathrm{CR}}$ frames on data channel.
- When CRU pair noted there is PU wants to claim the data channel.

For the first situation, the CRU pair stop transmitting/receiving data frames and need to hop back to control channel. The design of proactive channel vacating is focus on the second situation. The authors provide an interruption mechanism and elaborate it by three cases as follow.

After a successful **data transmission round** (the interval between RTS/CTS exchange and QP), a CR sender sends a RTI frame and then waits for SIFS_{CR} long. During this period, the CRU may hear a RTS from a PU who want to claim the channel. Due to the characteristic of wireless network, there are three cases while a PU sends out a RTS:

- Only CR sender overhears the RTS: the CR sender then stops transmitting data and hops back to the control channel. In the meantime, the CR receiver does not receive further data frames, and thus is aware that the CR sender must be interrupted by a potential primary sender. The CR receiver hops back to the control channel, too. If the CR sender still has data for the CR receiver, it generates a new initial data channel ID and increment-per-hop, and sends communication invitation on the control channel.
- Only CR receiver overhears the RTS: at this time, the CR receiver is aware of the presence of a primary sender. The CR receiver hops back to the control channel. Though the CR sender keeps sending data frames, it does not receive any ACKs from the CR receiver. Therefore, the CR sender vacates the occupied data channel and hops back to the control channel, too.
- Both CR sender and CR receiver overhear: both CR sender and CR receiver noted the presence of PU, and hop back to control channel.

With the design of proactive channel vacating mechanism, the CRUs can reduce the interference to PUs. After all, not interference PUs is one of the main sprit of cognitive radio.

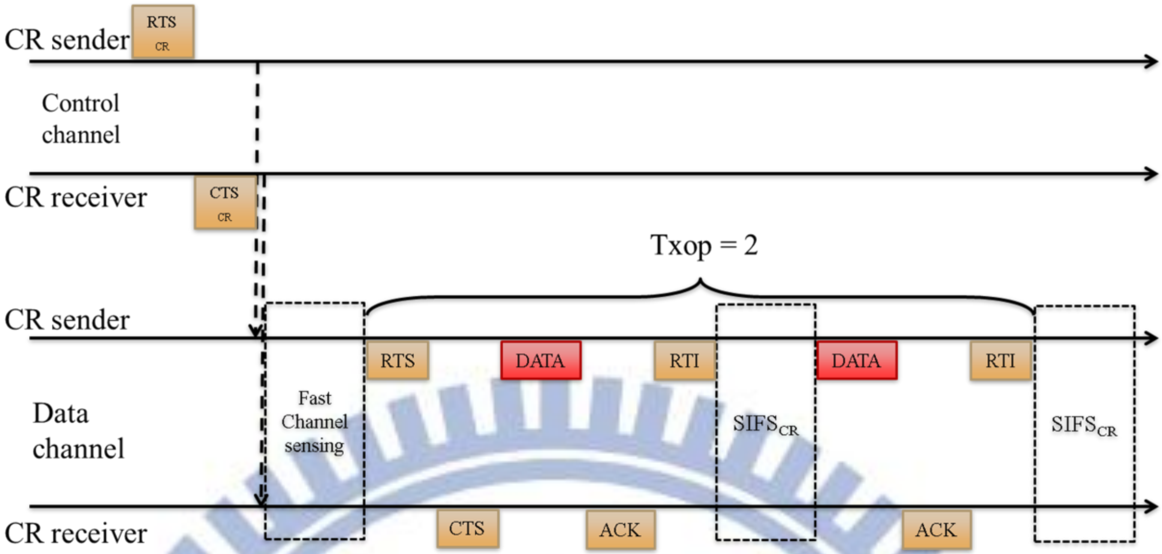


Figure 3.1 Illustration of Uni-MAC protocol with TXOP$_{CR}$ value is 2

- The drawback of Ori-MAC CR protocol

There are two drawbacks in the Ori-MAC CR protocol: (1) the inefficient channel access and (2) hidden terminal problem.

The channel hopping sequence of Ori-MAC is decided randomly that barely consider the actual channel status. The CRUs may switch to a data channel which is always occupied by PUs again and again. For example, if there is one data channel being always idle when other channels are occupied by PUs all the time. The Ori-MAC protocol needs to switch N-1 time to find the only available channel in the worst case. Each attempt to access an occupied channel wastes 2ms sensing time, that strongly decreases the efficiency of the protocol. Thus, Ori-MAC protocol is not efficient in finding an available channel under some situation.

Ori-MAC also omitted the hidden terminal problem which commonly exists in wireless network. Figure 3.2 illustrates the situation that hidden terminal problem might accrue. In this situation, CR sender sends the broadcast RTI and waits for SIFS$_{CR}$ after transmit the first data frames successfully. At this moment, PU sender start to claim the channel by sending the RTS which CR sender cannot hear. After waiting SIFS$_{CR}$ time, CR sender still considers the channel is idle and starts to transmit the data frame. This will cause that following frame transmitted by CR sender to collide with those transmitted by PU sender at the position of CR Receiver. The network flow goodput is greatly affected and the efficiency of Ori-MAC will greatly decrease when hidden terminal existed.

The two drawbacks were improved by the work of Shue-Yuan Wang et al., who proposed a enhanced CR MAC protocol introduced in the following section.

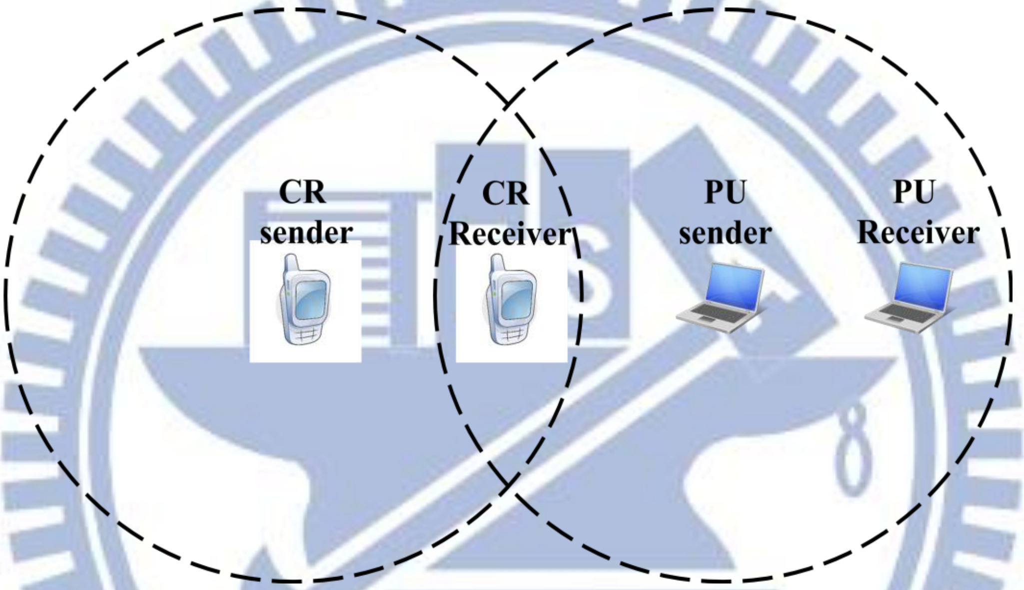


Figure 3.2 Illustration of Hidden Terminal Problem

3.2. Related work II: Enhanced MAC Protocol for Cognitive Radios over IEEE 802.11 Networks (Uni-MAC)

To solve the problems we mentioned in the previous section, Shue-Yuan Wang et al. [2] proposed two improvements to [1]'s Uni-MAC: *Smart Channel Selection Scheme* (SCSS) and *enhanced data transmission and channel evacuation process*. The detail of the two improvements is stated as follow.

- Smart Channel Selection Scheme

Shue-Yuan Wang et al. use SCSS to replace function (3-1) in [1] to generate the CRU’s hopping sequence. The SCSS can help CRUs to find the available channel in a better way. There are two phases in SCSS: *channel estimation* and *fast channel sensing*.

The *channel estimation* mechanism helps CRUs to choose the most possibly idle channel by using a 32-bit *Availability Record* (AR). Each CRU maintains a 32-bit AR for each data channel to record the historical channel availability result. CRU updates AR after each channel sensing opportunity such as fast channel sensing and QP. Before updating an AR, the CRU right-shift the value of the AR by one bit and record the most recently result at the most significant bit (MSB) of AR. The MSB of AR is set to 1 if the sensing result is idle, set to 0 if the sensing result is busy. The AR update process is shown in Figure 3.3.

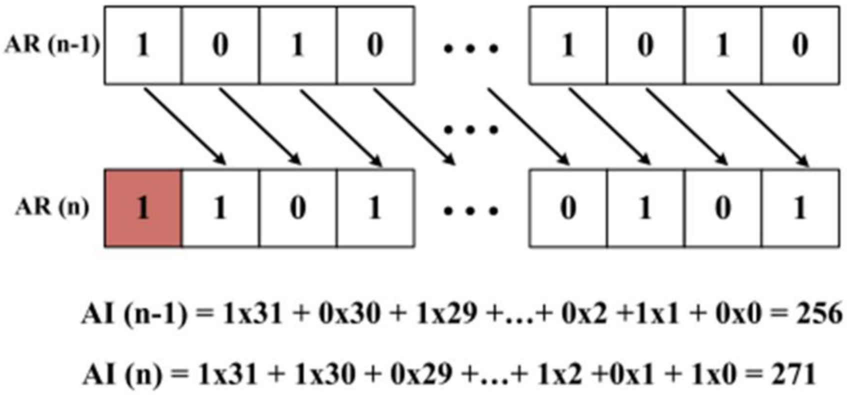


Figure 3.3 Illustration of AR update procedure and Calculation of the Availability Index

The Uni-MAC considers the channel is **IDLE** if it satisfy the following rules:

- Overhearing the ACK frame which means a data frame transmission is just finish.
- Not overhearing any MAC-layer frames which means the channel is unused.

The Uni-MAC considers the channel is B**USY** if it satisfies the following rules:

- Overhearing a RTS or CTS frame which means a PU or CRU is claiming the channel.
- Overhearing a data frame which means the channel is using by a PU or CRU.

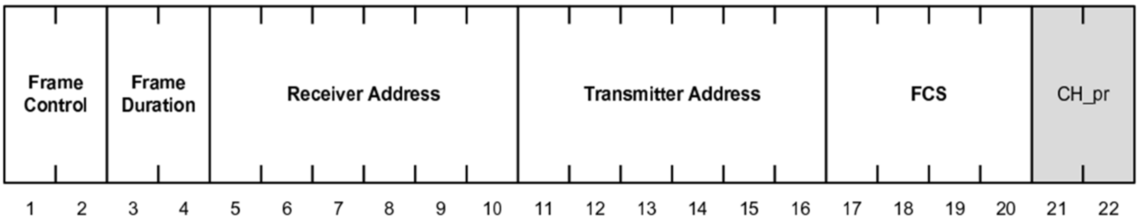


Figure 3.4 Frame format of RTS_{CR} in Uni-AMC

Figure 3.5 Frame format of CTS_{CR} in Uni-AMC

The new frame format of RTS_{CR} and CTS_{CR} of Uni-MAC is shown in Figure 3.4 and Figure 3.5. The initial selected data channel ID and increment-per-hop in RTS_{CR} of Ori-MAC is replaced by “channel_bitmap” field and a new “channel_hop_order_field” is add into CTS_{CR} frame. The definition of the two fields is elaborate as follow:

- channel_bitmap: a 16-bit long field which carries a candidate channels list from CR sender to inform CR receiver.
- channel_hop_order_field: a 32-bit long field which carry the channel hopping sequence from CR receiver replies to CR sender.

The candidate channels list is decided by an Availability Index (AI) for each data channel. AI is calculated according to AR, using the following equation:

$$\mathrm{AI} = \sum_{i=0}^{31} b(i) * pos(i) \tag{3-2}$$

which *i* is the index of AR's bit. The index of MSB is 31and the index of LSB (least significant bit) is 0. The index of other bits is shown in Figure 3.3. *b(i)* is the value of i^{th} bit and *pos(i)* is equal to the index value of the bit. There are two AI calculation examples in Figure 3.3 . The bigger AI value means the channel is not often used lately and a better selection for CRU to transmit data.

In Uni-MAC, when a CR sender has data to send, it also performs the *fast channel sensing process*. CRU selects the candidate channels according to AI and fills in the candidate list into "channel_bitmap" field of $\mathrm{RTS}_{\mathrm{CR}}$. CR sender then sends out the $\mathrm{RTS}_{\mathrm{CR}}$ frame to CR receiver. After receiving the $\mathrm{RTS}_{\mathrm{CR}}$, if CR receiver is available to receive data, it will do the *fast channel sensing* and reply the channel hopping sequence by "channel_hop_order_field" in $\mathrm{CTS}_{\mathrm{CR}}$. In the *fast channel sensing* process, the CR receiver switches itself to each selected data channel, and sense the availability status on that data channel for 100 microseconds. After sensing all selected data channels, it first updates its own AI values for the selected data channels and then determines the best order of channel hopping sequence among the selected data channels.

After $\mathrm{RTS}_{\mathrm{CR}}$/$\mathrm{CTS}_{\mathrm{CR}}$ handshake, the CR pair will hop into the data channel according the channel hopping sequence decided by AI and fast channel sensing that improve the efficiency by the simulation result of [2].

- Enhanced data transmission and channel evacuation process:

The reason of hidden terminal problem of Ori-MAC is because it only performs the RTS/CTS handshake procedure before the first data transmission round. However, the CR sender can transmit up to $TXOP_{CR}$ data frame after RTS/CTS handshake. During the rest of the CR transmission round, CR pair waits $SIFS_{CR}$ time for QP between two data transmission round, that providing opportunity for PU to claim the channel. At the QP, if a PU, which is hidden from CR sender, sends a RTS to claim the data channel, the RTS will not be overheard by the CR sender. After QP, CR sender still considers the channel is unused and sends out the data frame to CR receiver. As a result, a data frame collision happen between the CR sender's data frame and the data frame from PU at CR receiver.

To solve this problem, Uni-MAC performs RTS/CTS handshake procedure before any data transmission round. After QP, CR sender sends a RTS before data transmission. CR receiver will not reply to CR sender's RTS if it receives a PU (hidden from CR sender) sender's RTS. Without the reply from CR receiver, CR sender will learn the existence of PUs and hop back to control channel. This enhancement can easily solve the hidden terminal problem. Although the additional RTS/CTS handshake may increases the transmission overhead, it effectively prevents the interference between CRUs and PUs thus significantly increasing the goodput of the network when HTs are present. The enhancement of the Ori-MAC protocol is illustrated in Figure 3.6.

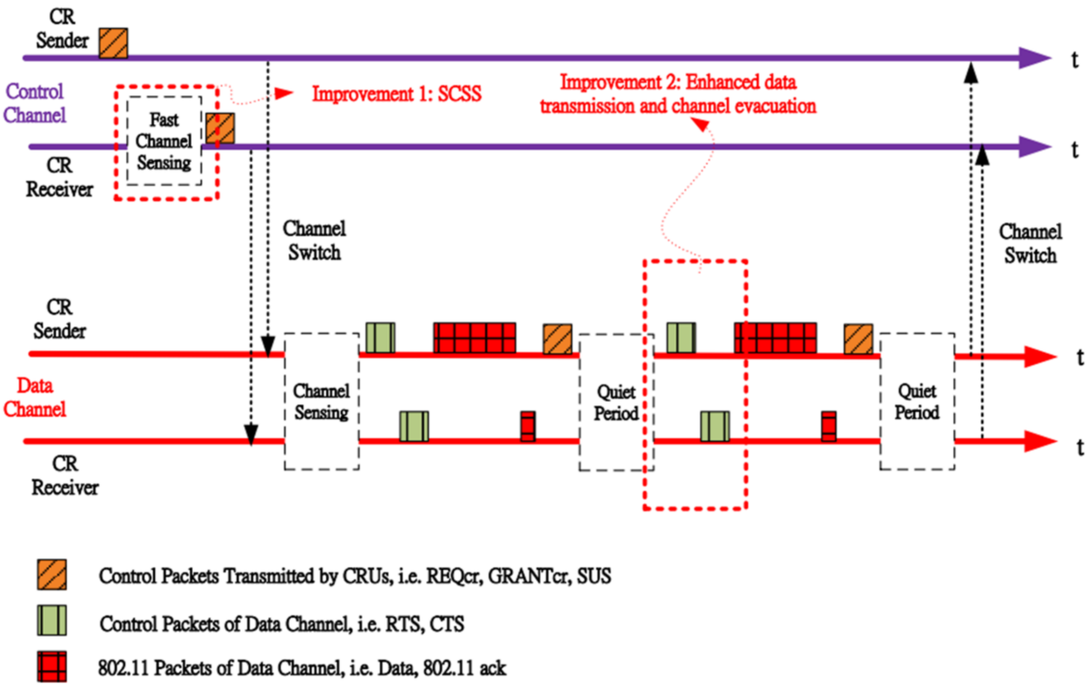


Figure 3.6 Illustration of Uni-MAC and the enhancement when TXOP$_{CR}$ value is 2

- The problem of Uni-MAC CR protocol

The two improvements of Uni-MAC do increase the throughput and efficiency of Ori-MAC. However, the simulation result in [2] shows that, compare to UDP flow's throughput the result of TCP flow's throughput is not good enough. The reason of the low efficiency of TCP flow and the solution is proposed by Lee-Chin Lau et al. in [3] which will introduce in next section.

3.3. Related work III: Bi-directional Cognitive Radio MAC Protocol for Supporting TCP Flows (Bi-MAC)

Lee-Chin Lau et al. identified that the major cause of the TCP performance degradation of Uni-MAC protocol that is due to the mechanism of TCP congestion control. This mechanism is employed such that a TCP sender would adjust its packet transmission rate based on the received TCP ACK. The authors propose a Bi-directional Cognitive Radio MAC Protocol (Bi-MAC) which is an extension of the CR MAC protocol proposed by [2]. The point of Bi-MAC is that it can smartly supports either uni-directionally or bi-directionally bandwidth reservation. The enhancements give the better supports of both TCP and two way UDP flows.

We will address the reason of how TCP congestion control influences TCP flow of Uni-MAC protocol in detail and how does Bi-MAC solve this problem in following.

- The degradation of TCP throughput of Uni-MAC

Consider a CR network without any PUs and only one CRU pair is transmitting a TCP flow. Let $\mathrm{CRU_A}$ be TCP sender that transmits the TCP packets continuously. Let $\mathrm{CRU_B}$ be the TCP receiver which replies TCP ACKs while receiving TCP packets from $\mathrm{CRU_A}$. The $\mathrm{TXOP_{CR}}$ in this case is set to 2.

For the sake of simplicity, we ignore the TCP three way handshake procedures at the beginning of TCP connection between the CR pair. When $\mathrm{CRU_A}$ start to transmit the TCP packets, it starts the BNP and sends a $\mathrm{RTS_{CR}}$ with the selected channel list to $\mathrm{CRU_B}$. After receiving $\mathrm{RTS_{CR}}$, $\mathrm{CRU_B}$ performs a 500 us *fast channel sensing* (100 us pre data channel), updates AR and replies the selected channel order to $\mathrm{CRU_A}$. Then, CRU pair hops to first selected data channel and start channel sensing for 2 ms. After that, $\mathrm{CRU_A}$ can transmit two frames with a 100 us QP after receives the replied 802.11 ACK from $\mathrm{CRU_B}$. After receiving the first TCP packet from $\mathrm{CRU_A}$, $\mathrm{CRU_B}$ starts to generate the TCP ACK. However, with the design of Uni-MAC protocol, $\mathrm{CRU_B}$ cannot reply the TCP ACK to $\mathrm{CRU_A}$ at the first moment. In order to reply TCP ACK to $\mathrm{CRU_A}$, $\mathrm{CRU_B}$ has to wait for the current CR transmission round finish and start over from the BNP again. After the BNP, the CRU pair should find an available data channel before $\mathrm{CRU_B}$ can reply TCP ACK to $\mathrm{CRU_A}$. Finally, $\mathrm{CRU_B}$ can transmit TCP ACK on an idle data channel.

Figure 3.7 illustrates the situation mention above. One can see there are two drawbacks for TCP flow due to the one-way design of Uni-MAC.

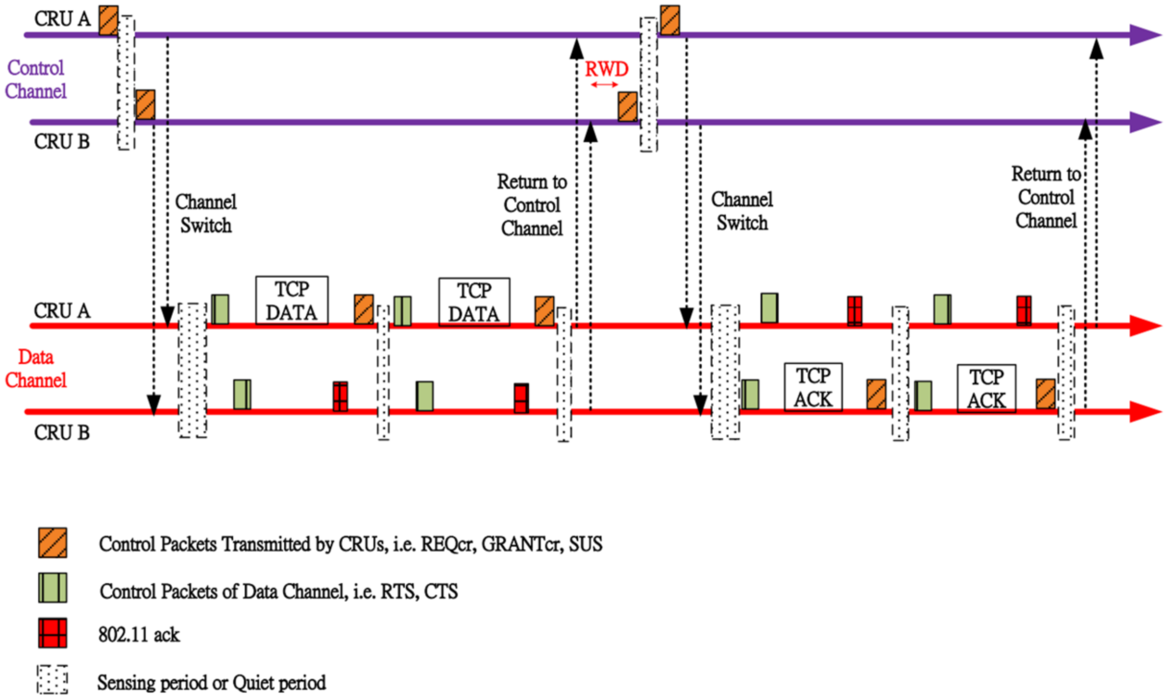


Figure 3.7 Illustration of the best case of the TCP packets exchanges when TXOP$_{\text{CR}}$ is 2

First, CRU$_{\text{B}}$ cannot reply TCP ACK immediately after it receives a TCP packet. CRU$_{\text{B}}$ has to wait until CRU$_{\text{A}}$'s CR transmission round finishes and contains with other CRU for the BNP in control channel. In control channel, CRUs use IEEE 802.11 DCF [7] to access the channel. If there are many other CRUs attempt to start BNP, the CRU$_{\text{B}}$ has to contain with them and may not get the opportunity to send the RTS$_{\text{CR}}$ right after the CRU$_{\text{A}}$'s CR transmission round finishes. After BNP, the CRU pair still needs to find an available data channel through *fast channel sensing*. Although Uni-MAC implements the SCSS which increase the probability for CRU pair to find the available data channel at the first try; it still has a chance to fail and need a second or even a third try. **The process mentioned above decided the round-trip time (RTT) of the TCP flow.** In the best case (see Figure 3.7), with no contention and failure during the BNP and SCSS, the RTT time is the summation of data transmission time, QP time, time of BNP and channel sensing time. However, in a bad case (see Figure 3.8), the SCSS may not success at the first try and will costs additional BNP time or channel sensing time. This highly increases the RTT of the TCP flow of CR sender and decrease the performance.

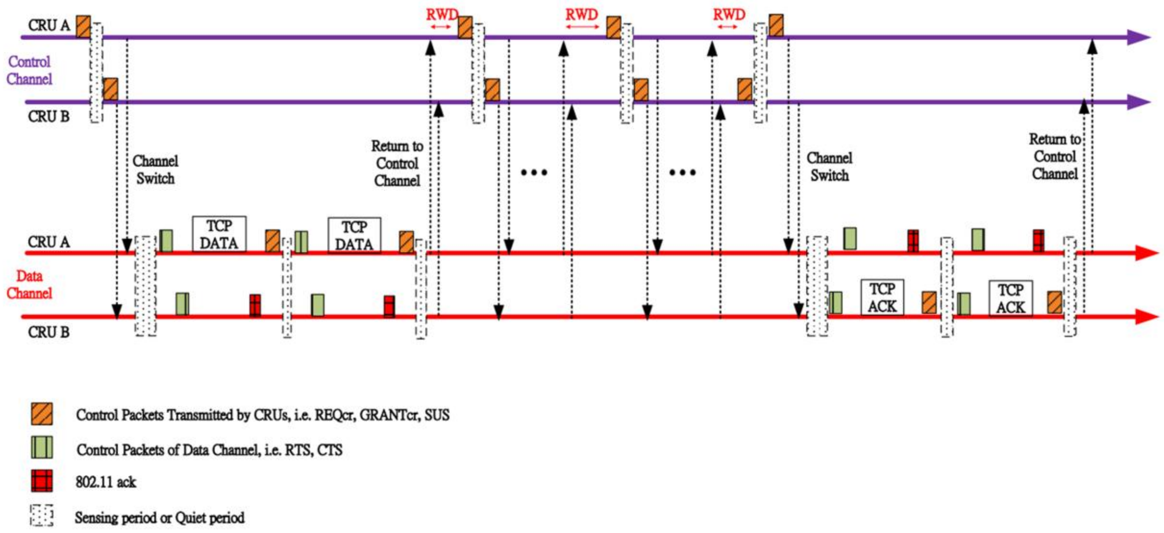


Figure 3.8 Illustration of a bad case of the TCP packets exchanges when TXOP$_{CR}$ is 2

Second, if the receiver’s TCP ACKs arrive at the sender with the same spacing, then the sender can send new data packets at the same rate to avoid overrunning the bottleneck link. It is said that such ACK policy makes the protocol self-clocking because the sender can dynamically adapts its transmission speed to both the speed of the network and the speed of the peer sending TCP ACKs. Since a TCP’s self-clocking depends on the arrival of TCP ACKs at the same spacing with which the receiver generated them, if these TCP ACKs spend any time sitting in queues during their transit through the network, their spacing may be altered. When ACKs arrive closer together than they were sent, the sender might be misled into sending more data than the network can accept, which could led to congestion and loss of eficiency. This is called the ack-compression effect. It has been proven both statistically and experimentally that the ack-compression effect could result in unfairness and reduced overall throughput compared to what could be expected without this effect.

- Proposed Bi-MAC protocol in [3]

To solve the problem, the authors propose a design which supports both bi-direction traffic and uni-direction traffic. A 2-bit field is added into RTS_{CR} and CTS_{CR} control frame[1]. The extension filed name as *Reservation Type* (RT), which indicates the bandwidth reservation type for CRU pair in a transmission run. In Bi-MAC, before a CRU transmitting RTS_{CR} or CTS_{CR}, a CRU always checks the head of transmitting queue (HOQ) and set RT in following rules:

- CR sender set RT of RTS_{CR} to *00* if CR sender identifies HOQ is an UDP packet. This indicates CR sender is issuing a uni-direction UDP flow.
- CR sender set RT of RTS_{CR} to *01* if CR sender identifies HOQ is a TCP packet. This indicates CR sender is issuing a bi-direction TCP flow.
- CR receiver set RT of CTS_{CR} to *10* if CR receiver identifies HOQ is not empty. This indicates CR receiver requesting a bi-direction flow.
- CR receiver copies the RT value from received RTS_{CR} to the RT of CTS_{CR} if CR receiver identifies HOQ is empty. Because the queue of CRU's is empty, it follows the RT demanded reservation type by CR sender.

In summary, the bandwidth reservation of a CR transmission round is decided by the RT value of CTS_{CR}: *00* means uni-direction bandwidth reservation is demanded by CRU pair and bi-direction bandwidth reservation is demanded by CRU pair for other value.

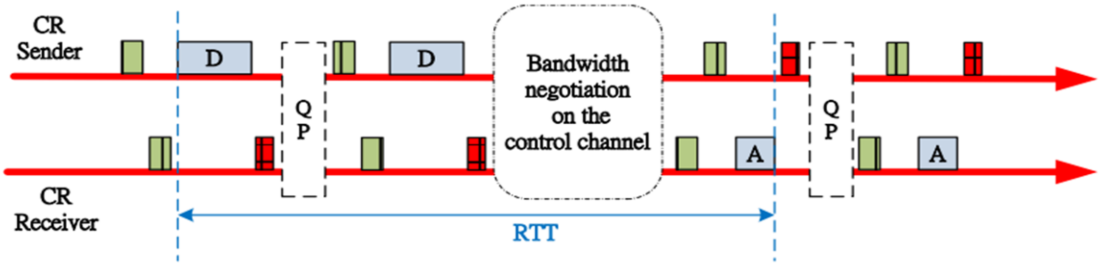
---

[1] In [3] and [4], the authors change RTS_{CR} and CTS_{CR} to REQ_{CR} and $GRANT_{CR}$. We change it back to RTS_{CR} and CTS_{CR} for consistence.

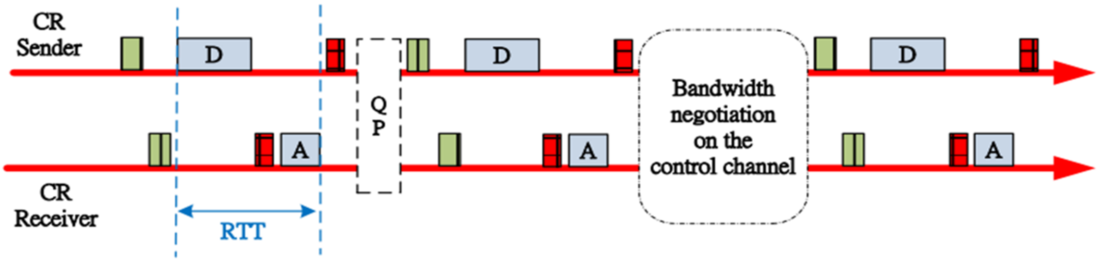
After BNP and channel sensing, the transaction interval (TI) to be filled in a RTS control packet is decided by the bandwidth reservation type. The authors only consider a simple network case with either a TCP flow or bi-directional UDP flows with uniform packet size between a CRU pair. With one-way bandwidth reservation, the TI is determined simply based on the packet length of the HOQ at the CR sender.

For a two-way bandwidth reservation, however, the TI is determined according to the value of RT of CTS_{CR}. If the value of RT of CTS_{CR} is filled with 01, a TCP flow is anticipated. Therefore, the TI will be set to a duration which is adequate for transmitting a TCP data frame, a TCP ACK and two 802.11 ACK packets. Alternatively, if the RT of CTS_{CR} is filled with a value of 10, two UDP flows of opposite directions are anticipated. In this case, the TI will be set to a duration which is adequate for transmitting two UDP packets of equal size and two 802.11 ACK packets.

With the design of two-way bandwidth reservation, the CR receiver can reply ACK right after the CR sender transmits the data frame. This can make RTT of TCP flow to a smaller value. In Figure 3.9, we illustrate the RTT of a TCP flow of both Uni-MAC and Bi-MAC. One can see the big improvement for reducing the RTT with the proposed two-way bandwidth reservation of BI-MAC. The simulation results in [3] also prove that the proposed Bi-MAC protocol can dramatically improve both the TCP and UDP throughput of CRUs without sacrificing the performance of PUs.



a) Uni-MAC (Txop = 2)



b) BBi-MAC (Txop = 2)

Figure 3.9 Illustration of Packets Exchanges on Data Channel of a) Uni-MAC and b) Bi-MAC, when $TXOP_{CR}$ is 2

3.4. Related work IV: An IEEE 802.11 cognitive radio MAC protocol with dynamic bandwidth allocation capabilities.

In former section, Lee-Chin Lau et al. propose a bi-directional CR MAC protocol which solves the TCP low efficiency problem of Uni-MAC protocol. However, the design of Bi-MAC protocol is inadequate in supporting traffic demands in more complicated network cases. To solve the problem, the authors propose an Advance-Bi-direction CR MAC protocol (ABi-MAC) with *Dynamic Bandwidth Allocation* (DBA) algorithm to supports asymmetric two-way traffic flows. Furthermore, the proposed *Smart Transaction Interval Setting* (STIS) guarantees the Quality of Service of PUs. In this section, we first address the problems of Bi-MAC and then introduce the solution of ABi-MAC.

- The definition of $TXOP_{CR}$ in Uni-MAC and Bi-MAC.

In Uni-MAC and Bi-MAC, $TXOP_{CR}$ is adopted as a fixed network parameter. In Uni-MAC, the $TXOP_{CR}$ is the number of maximum data frames that could be transmit by CR sender after CRU pair in a CR transmission round. In Bi-MAC, if two-way bandwidth reservation is adopted, $TXOP_{CR}$ is the number of maximum data frames that could be transmit by both CR sender and CR receiver. One should note that, during the CR transmission round when two-way bandwidth reservation is applied, each of the CR pair can send a data frame that costs two $TXOP_{CR}$ in a data transmission round. Otherwise, the meaning of $TXOP_{CR}$ is the same as Uni-MAC.

A small value of $TXOP_{CR}$, i.e. one, should be used when the data channels is usually in a busy state. However, if data channel is usually in an idle state, a huge overhead is caused by the time of BNP and SCSS after each data transmission run. In general, a large value of $TXOP_{CR}$, i.e. four, is suitable for all network condition. If the data channel is usually idle, CRUs can stay a longer time on data channel for possible data transmission. On the other hand, when data channels are busy, the CRU still guarantee the QoS of PUs' data transmission. Due to the *channel vacating mechanism*, the CRU will vacate the data channel and hop back to control channel when a PU tries to claim the channel at QP. However, for a large value of $TXOP_{CR}$, i.e. four, will cause the unavoidable overhead when bi-directional bandwidth reservation is considered. The problem is address as following.

- The unavoidable overhead of large $TXOP_{CR}$ with bi-directional bandwidth reservation of Bi-MAC

The fixed value of TXOP_{CR} causes the bandwidth waste while a CRU pair is transmitting asymmetric two-way traffic flows. Figure 3.10 illustrates the overhead while CRU is using Bi-MAC with bi-directional bandwidth reservation when TXOP_{CR} equal to 3. One can see that CR receiver's queue is empty after the first data transmission turn. In this case, the TXOP_{CR} is set to three and there is no PUs try to claim the channel during QP. In the second data transmission round, CR sender transmits the data frame and not receiving the data frame from CR receiver. Though, the CRU pair enter QP after the timer of the bandwidth reservation expires. Again, the situation remains the same in the third data transmission turn. The CRU pair still waste their time until the timer expire after CR sender transmits the data frame. As a result, the time spent waiting for the timer to expire is an unavoidable overhead of Bi-MAC while TXOP_{CR} is fixed. The worst case is that the queue of CR sender also becomes empty before the TXOP_{CR} transmission turn finish. In this situation, the CRU pair occupy the channel and do nothing until the timer expire. Such undesirable phenomena not only increase the overhead, but could also influence the PUs performance. The reason of the influence to PUs is due to the fixed TI setting which is explained as follow:

- The problem of fixed TI setting with two-way bandwidth reservation of Bi-MAC.

Bi-MAC is designed to support one TCP flow between CRU pair or two way UDP flow with uniform same size between CRU pair. The TI setting of RTS control frame with two-way bandwidth reservation is determined on the RT value of CTS_{CR}, as elaborated in section 3.3. However, the design does not satisfy the situation of two-way asymmetry UDP traffic over a CRU pair.

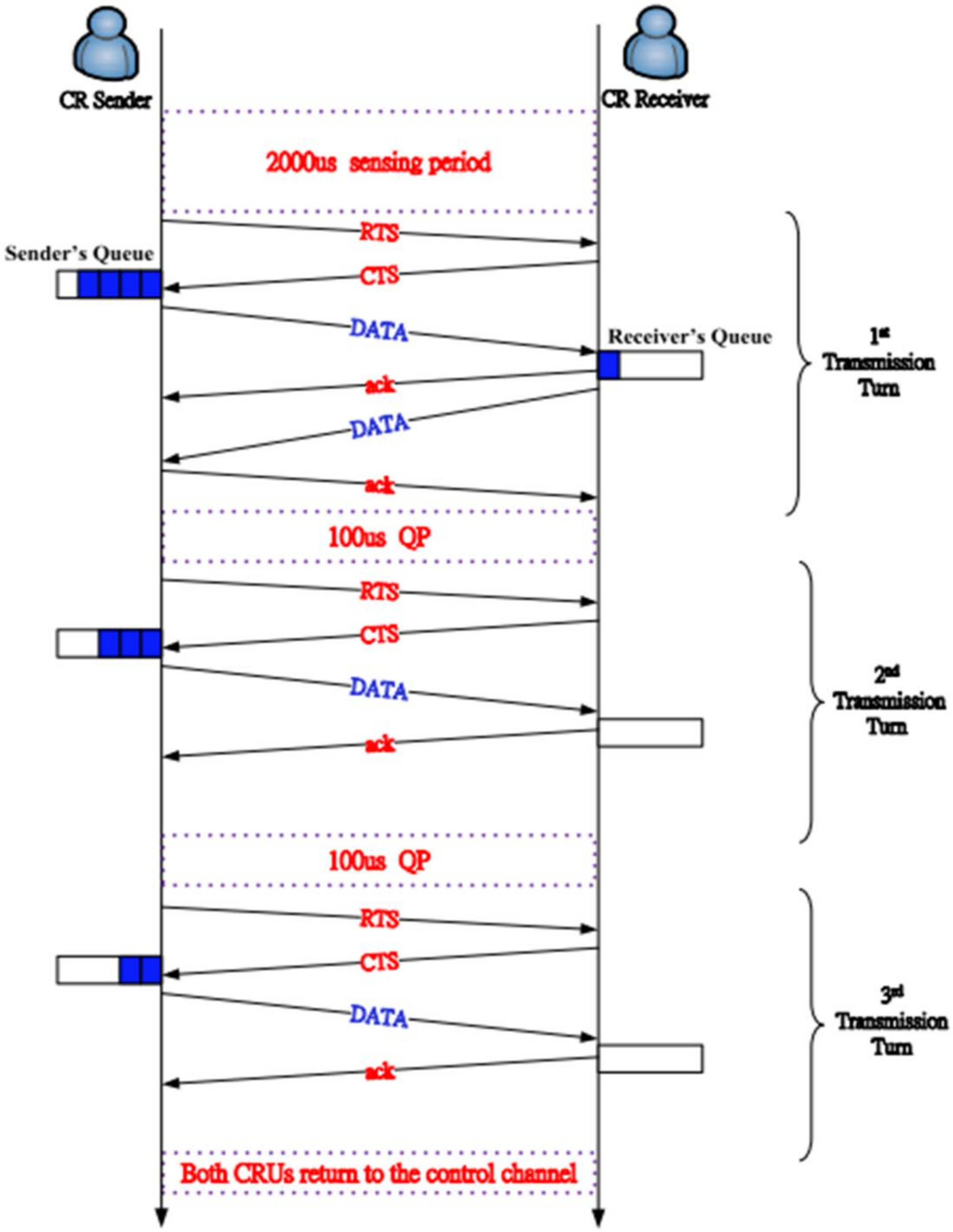


Figure 3.10 Illustration of Message Exchanges on Data Channel of Bi-MAC with Bi-directional Bandwidth Reservation when $TXOP_{CR} = 3$

Figure 3.11 illustrates the negative impact to PUs QoS which is caused by the fixed TI setting in Bi-MAC when two-way bandwidth reservation is considered. In this case, the size of data frames in CR sender’s queue is 1,450 bytes. On the other hand, the there is only one 200 bytes long data frame in CR receiver’s queue. The transmission run perform two-way bandwidth reservation due to the non-emptiness of CR receiver’s HOQ. With the design of Bi-MAC, TI is set to a duration which is long enough for transmitting two 1,450 bytes long data frames and two 802.11 ACK frames.

One can see in Figure 3.11, the first TI set to the RTS is longer than the actual transmission time of the first data transmission round due to the small packet of CR receiver. This inappropriate design strongly breaks the opportunity of PUs to claim the channel. The CRU pair entered the QP right after the ACK reply to CR receiver is transmitted. However, due to the RTS transmitted at the beginning of data transmission round, all neighboring PUs will set their Network Allocation Vector (NAV) according to TI of RTS. The NAV is an indicator for a station on how long it must defer from accessing the medium. In this case, the neighboring PUs receives the RTS will defer until the bandwidth reservation time of CRUs is finished. As a result, the CRU pair enter QP in the reserved time which PUs cannot send a RTS to claim the channel due to NAV. After QP, CRU starts the second data transmission round and transmits RTS again. The neighboring PUs overhear the RTS and set their NAV according to TI again. At this moment, one can see the reserved time of CRU pair of first transmission run is not finish yet due to the small size frame of CR receiver. The reserved time of CRU pair of second transmission round overlaps with the first one that gives no free time for PUs to claim the channel in between the first data transmission round and second transmission round. The same case happens when the CR receiver is running out of frames in its queue, as shown in the CR receiver’s second transmission round in the figure. In this case, both the CRUs enter the QP after the timers expire but the PUs are still deferring from accessing the medium due to the NAV.

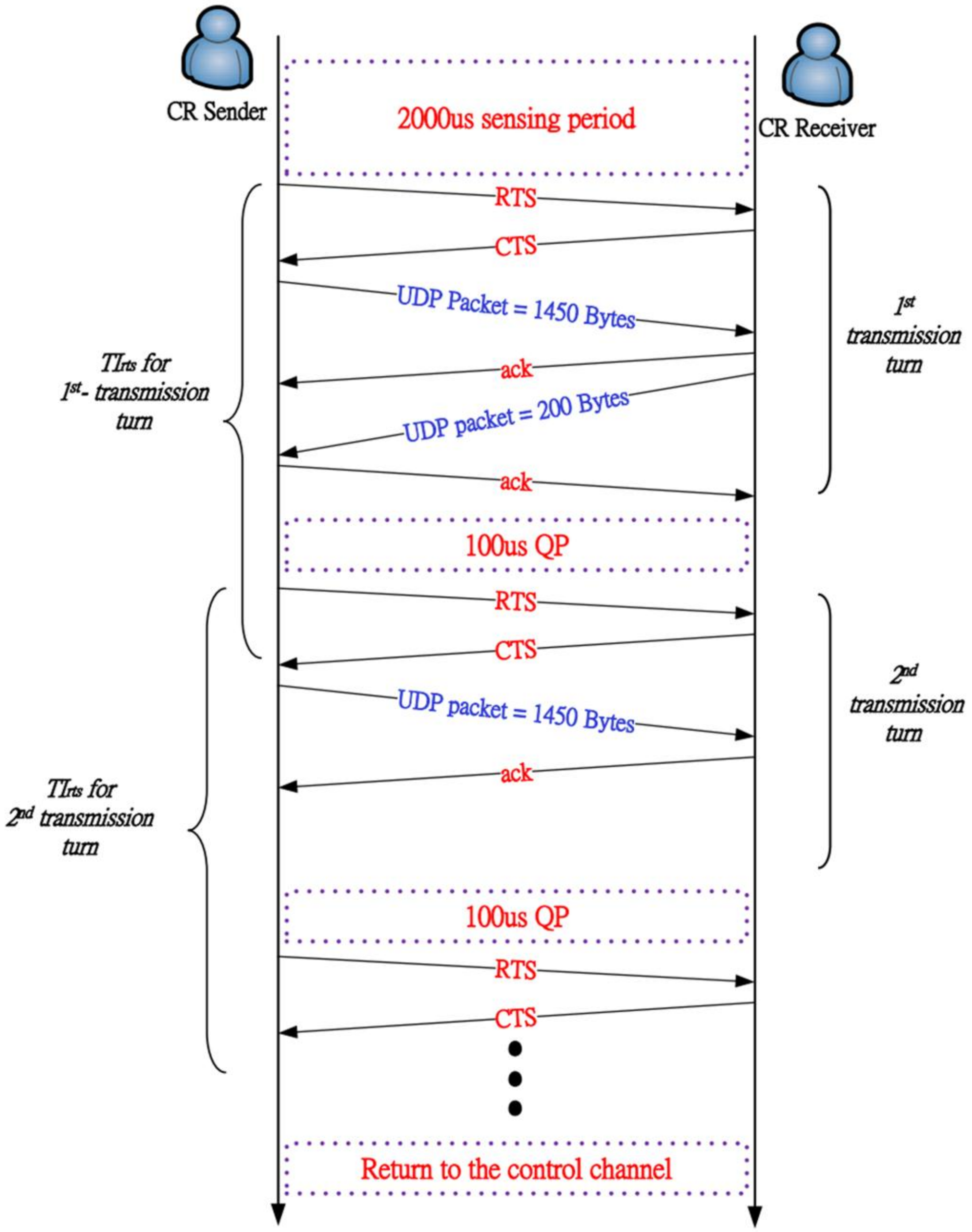


Figure 3.11 Illustration of inappropriate TI setting of Bi-MAC with Bi-directional Bandwidth Reservation

The CRU pair elaborated above do enter the QP after every transmission round. However, the result is it does not protect the PUs’ right to claim the channel at all. In other word, the QP is totally a waste of time. Furthermore, the CRU pair occupy the data channel until the end of the $TXOP_{CRth}$ data transmission round. This does not satisfy the spirit of cognitive radio: the CRU should not interferes the right of primary users.

- The design of Advance-Bi-direction CR MAC protocol (ABi-MAC)

To solve the above two problems, the authors of [4] propose ABi-MAC with two phases as follow:

Dynamic Bandwidth Allocation **(DBA)**

The main purpose of DBA is to determine the available transmitting frame number for both CR sender/receiver right after the BNP. Thus, the CRU pair can use the channel in a more efficient way to transmit the data frames on data channel. Through this purpose, the authors defined several new variations of TXOP:

- MAX_PACKET: a pre-defined parameter. The maximum number of frames that CRU pair can transmit on the data channel. We use $TXOP_{CR}$ instate in this paper
- BD_s: the bandwidth demand of CR sender. It is a 5-bit field at the end of RTS_{CR} control frame. Representing the bandwidth demand from 0~31 data frames.
- BD_r: the bandwidth demand of CR receiver. It is a 5-bit field at the end of CTS_{CR} control frame. Representing the bandwidth demand from 0~31 data frames

Both BD_s and BD_r are decided dynamically through the BNP. In ABi-MAC, BD_s and BD_r is carried in the RTS_{CR} and CTS_{CR} control packet separately. The *Reservation Type* (RT) in Bi-MAC is replaced with a 5-bit field of BD_s or BD_r, as shown in Figure 3.12 and Figure 3.13.

Figure 3.12 Frame Format of the control packet RTS_{CR} of ABi-MAC

Figure 3.13 Frame Format of the control packet CTS_{CR} of ABi-MAC

The DBA of ABi-MAC operates in the following way. Firstly, assume that a CR sender maintains a separate queue for every CR receiver. Before sending an RTS_{CR}, the CR sender gets the number of data frames to be transmitted to a specified CR receiver by retrieving the current size of the corresponding queue. This information is treated as the value of BD_{s} in the RTS_{CR} and shall be set using the following equation:

$$BD_s = \begin{cases} Q_s, & \text{if } Q_s \leq M \\ M, & \text{if } Q_s > M \end{cases} \qquad \text{(3-3)}$$

where Q_{s} denotes the current queue size of the CR sender (for the specified CR receiver) and M denotes the pre-defined and fixed network parameter TXOP_{CR}. Upon receiving the RTS_{CR}, the CR receiver checks the corresponding queue that contains packets to be sent to the CR sender and then decides the value of BD_{r} using Eq. (3-4). At the same time, the CR receiver also records the value of BD_{S} as the number of packets to be received from the sender (NPR_{S}) using Eq. (3-5). After that, it replies a CTS_{CR} to the CR sender.

$$BD_r = \begin{cases} 0, & \text{if } Q_r = 0 \\ M - BD_s, & \text{if } Q_r \geq \lfloor A \rfloor \text{ and } BD_s \leq \lceil A \rceil \\ \lfloor A \rfloor, & \text{if } Q_r \geq \lfloor A \rfloor \text{ and } BD_s \geq \lceil A \rceil \\ Q_r, & \text{if } Q_r \leq \lfloor A \rfloor \end{cases} \qquad \text{(3-4)}$$

where Q_r denotes the size of the queue containing packets to be sent to the CR sender and A denotes M/2.

$$NPR_s = \begin{cases} BD_s, & \text{if } BD_r = 0 \\ M - BD_r, & \text{if } BD_r > 0 \end{cases} \tag{3-5}$$

Finally, the CR sender reads the value of BD_r in CTS_{CR} and records the number of packets to be received from the receiver (NPR_r) using Eq. (3-6). It then finalizes its value of BD_s using the Eq. (3-7) where BD_{SF} denotes the finalized value of BD_s at the end of the BNP.

$$NPR_r = \begin{cases} BD_s, & \text{if } BD_r = 0 \\ M - BD_r, & \text{if } BD_r > 0 \end{cases} \tag{3-6}$$

$$BD_{SF} = \begin{cases} BD_s, & \text{if } BD_r = 0 \\ M - BD_r, & \text{if } BD_r \leq \lfloor A \rfloor \text{ and } BD_s \geq \lceil A \rceil \\ BD_s, & \text{if } BD_r \geq \lfloor A \rfloor \text{ and } BD_s \leq \lceil A \rceil \\ BD_s, & \text{if } BD_r \leq \lfloor A \rfloor \text{ and } BD_s \leq \lceil A \rceil \end{cases} \tag{3-7}$$

Using the proposed DBA, the ABi-MAC can offer very high flexibility in allocating bandwidth to the CRU pair regardless of their traffic condition. Through maintaining the BD_s, BD_r, NPR_s and NPR_r, ABi-MAC can support (1) a uni-directional traffic flow from the CR sender to the CR receiver, (2) a bi-directional traffic flow where both the CRU sender and the CR receiver have the same number of packets to send in their queues, and (3) a bi-directional traffic flow where the CRU sender and the CR receiver have unequal numbers of packets to send in their queues. In case (3), the CRU pair will swap their sender/receiver identities on the data channel when the CR sender has transmitted all packets in its queue but the CR receiver still has packets to send in its queue. At this moment, the bi-directional traffic flow temporarily degenerates to a uni-directional traffic flow.

Smart Transaction Interval Setting **(STIS)**

In phase two of the ABi-MAC, we proposed STIS such that the TI to be carried in the control packets can provide the neighboring PUs with the correct timing information when bi-directional bandwidth reservation is needed. In a typical 802.11 network, a two-way handshake of RTS/CTS is adopted for bandwidth reservation. In the process of handshaking, the TI is carried in the RTS packet to indicate how long a sender wants to hold the medium. In return, the receiver replies with a CTS packet echoing the expected duration of transmission. Through the exchange of RTS and CTS control packets, all the nodes within the hearing distance of either the sender or receiver or both will set their Network Allocation Vector (NAV) according to the TI in the overheard packets.

In the presence of bi-directional traffic flow, we further proposed a conditional transmission of RTS_e by the CR sender after receiving a CTS. As a result, the CR sender can effectively update all of its neighboring PUs with its latest TI whenever bi-directional bandwidth reservation is required. The format of RTS e in ABi-MAC is identical to that of the RTS packet. In the STIS, the TI of a CTS packet is set according to the following equation:

$$TI_{cts} = TI_{rts} - sifs - D_{cts} + D_{rts_e} + 3sifs + D_{data} + D_{ack} \tag{3-8}$$

$$TI_{rts_e} = TI_{cts} - sifs - D_{rts_e} \tag{3-9}$$

where D_{cts} , D_{rtse} , D_{data} , and D_{ack} refers to the transmission time of a CTS, RTS_e , data, and ACK packet, respectively. Upon receiving a CTS packet, if a bi-directional bandwidth reservation is anticipated, the CR sender immediately transmits an RTS_e control packet with the TI_{rtse} calculated based on Eq. (3-9).

In summary, the proposed ABi-MAC can reserve bandwidth dynamically to further utilize the spectrum holes of the authorized bandwidth. In addition, the ABi-MAC can support asymmetric two-way traffic flows with variable packet sizes. Simulation results show that our proposed ABi-MAC can significantly enhance the aggregate throughputs of CRUs in a CRN and dramatically improve the spectrum efficiency without degrading the performance of PUs.

Chapter 4 Motivation

The related work in chapter 3 all focus on the same target: a CR MAC protocol over IEEE 802.11 network. Those works do make a significant contribution for CRN. However, the verification or simulation in [1], [2], [3], and [4] seems unable to satisfy the real world situation with the fast growing technology. In those works, they use IEEE 802.11b stations to be PUs that is barely used in these days. And the data rate applied in the verification and simulation for both PUs and CRUs is 2 Mbps that is also an unusual choice in real world. The new versions IEEE 802.11a/g/n are now widely used in the real world and the supported data rate of IEEE 802.11a/g is up to 54 Mbps. The IEEE 802.11n even supports a 150 Mbps high data rate with only one single antenna [8]. We think those work should be examined and verified under the conditions of using higher data rate and using a more advance version of IEEE 802.11 stations to be PUs.

Furthermore, we consider an improvement to enhance the efficiency of CR-MAC protocol through frame aggregation mechanism which is proposed in IEEE 802.11n [8]. Since, physical level data rate improvements do not increase user level throughput beyond a point because of 802.11 protocol overheads, like the contention process, inter frame spacing, physical level headers (Preamble + PLCP) and acknowledgment frames. The same problems should happen in CR MAC protocol and could be more serious due to other protection concern for PUs like channel sensing and QP overhead. The frame aggregation mechanism is proved that it does improve the efficiency of IEEE 802.11 (since IEEE 802.11n is widely used in the real world). We can expect that frame aggregation mechanism could also enhance the efficiency of CR MAC protocol. However, the CR MAC protocol has to consider the protection of PUs additionally. In the following chapter, we will implement the frame aggregation mechanism on both Uni-MAC and ABi-MAC to increase the available throughput and analysis the effect from the enhancement to PUs.

Chapter 5 Proposed Frame-aggregated CR Mac protocol (FA-MAC)

In this chapter, we will introduce the design and implementation of the proposed FA-MAC. FA-MAC is defined to known as CR MAC protocol applied with different frame aggregation mechanism. We only propose frame aggregation mechanism on Uni-MAC and ABi-MAC. The reason is that Uni-MAC can be seen as an advance version of Ori-MAC and ABi-MAC is the most complete version of all CR MACs in Chapter 3. Also, the difference between the Uni-MAC and ABi-MAC is the support of bi-direction traffic flow. We would like to see how does frame aggregation mechanism works on both Uni-MAC and ABi-MAC protocols. We also care about the effect of the protection to PUs from them.

In order to analysis two CR MAC protocols with higher data rate and new IEEE 802.11a PUs, we first implement the Uni-MAC and ABi-MAC modules on Estinet network simulator [14]. The Estinet network simulator is the commercial version of NCTU network simulator [15] (NCTUns) with more advance IEEE 802.11 wireless network such as IEEE 802.11a and IEEE 802.11n. The new version of IEEE 802.11 network could be the better PUs for the verification and simulation of CR MAC protocols.

During the verification of the correctness of Uni-MAC and ABi-MAC protocol on Estinet, we observed a defect of BNP which will increase the data transmission overhead. We propose a New BNP to reduce the bad impact. We also modified the Dynamic Bandwidth Allocation (DBA) into a more dynamic version. Both enhancements will address it in the following section. After that, we will introduce the implementation of frame aggregation and block ACK mechanism of CR MAC protocol.

5.1. Migration of Uni-MAC and ABi-MAC over Estinet Network Simulator

During the migration of Uni-MAC and ABi-MAC, we find some new improvement and change some parameter settings due to higher data rate is applied.

First, we rebuild Uni-MAC and ABi-MAC according to Estinet IEEE 802.11a module which has a more powerful physical layer and a more precise random back off scheme in MAC layer.

The random back off scheme is both implemented by a set of timers on NCTUns and Estinet. In IEEE 802.11 DCF (see Figure 5.1(a)), a station has to defer DIFS after channel is clear and random back off a random number of back off slot before accessing channel. Both Estinet IEEE 802.11a module and old NCTUns IEEE 802.11b module uses a timer to count the DIFS time. If the channel is sensed busy when a station is deferring DIFS time. The old scheme suspends the DIFS timer and starts the timer after the channel become idle again (see Figure 5.1(b)). However, in IEEE 802.11- 2007 [7], when a station is interrupted when deferring DIFS time, the station should count another complete DIFS time again. This is different from the old scheme that does not recount all DIFS time again. The same problem happens on the random back off process. In old scheme (see Figure 5.1(b)), it uses only one counter to count down the random back off time. The random back off operation is counted slot by slot in spec IEEE 802.11- 2007. When channel become busy again during random back off process, a station should stop at the counted slot and start over from the beginning of the slot again. The same problem happens due to the old scheme does not recount a slot again.

On the other hand, the new scheme proposed by Estinet works exactly the same how IEEE 802.11 DCF works in spec.

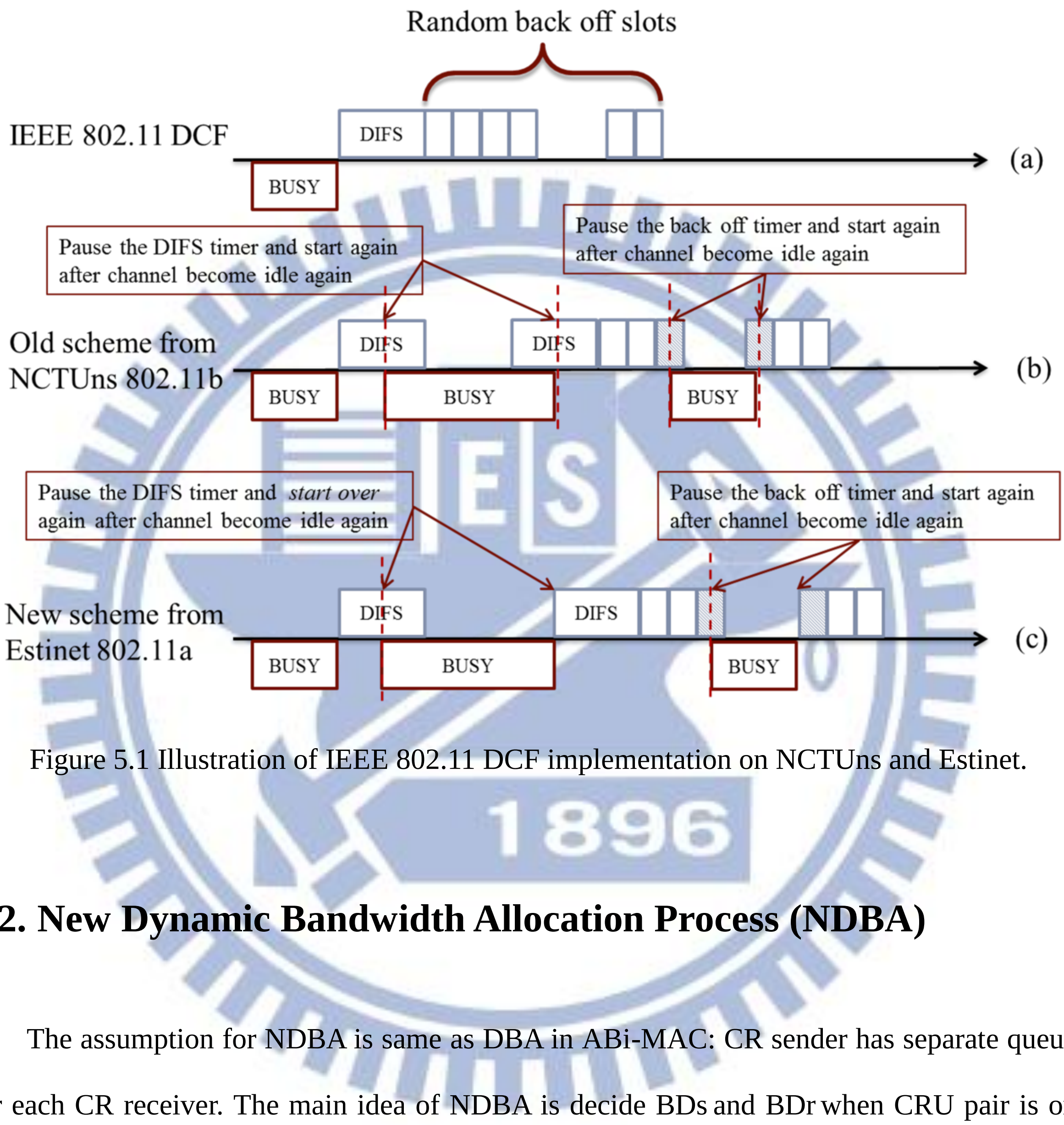


Figure 5.1 Illustration of IEEE 802.11 DCF implementation on NCTUns and Estinet.

5.2. New Dynamic Bandwidth Allocation Process (NDBA)

The assumption for NDBA is same as DBA in ABi-MAC: CR sender has separate queue for each CR receiver. The main idea of NDBA is decide BDs and BDr when CRU pair is on the data channel and BDs and BDr is decided at the beginning of each data transmission round. To achieve this goal, the CR pair exchange their queue length information during RTS/CTS handshake. We extend RTS and CTS control frame with a 4-bit field which contain the current queue length information. In addition, both CR sender and CR receiver maintain the remaining TXOP ($TXOP_R$) in order to know whether two-way bandwidth reservation is allowed. With the queue length information and remaining TXOP, NBDA is work as follow.

First the BD_s and BD_r is decided by the remaining $TXOP_{CR}$ value directly:

- If $TXOP_R$ is bigger than 1, the value of both BD_s and BD_r is decided by the queue length of CRU. If its queue length is not zero, BD_s/BD_r is equal to 1. If its queue empty, BD_S/BD_R is zero.
- If $TXOP_R$ is equal to 1, then BD_S is still equal to one and BD_R is zero.
- If $TXOP_R$ is zero, the CRU pair hop back to control after QP.

With $TXOP_R$, CR sender and CR receiver is easily to tell whether two-way bandwidth reservation is applied during this transmission. If two-way bandwidth reservation is applied, the CR receiver will automatically applied the *Smart Transmission Interval Scheme* (STIS) by extending the duration of CTS with its own frame transmission time and an ACK transmission time. CR sender will check the duration in CTS to know that if two-way bandwidth reservation is applied or not in this data transmission round. If it is applied, CR sender sends out RTS_e to inform the PUs in neighborhood for two-way bandwidth reservation.

Second, the CR sender and CR receiver can tell from the queue length information that is there any packet from each other after this transmission. Without this information, the CR sender will not change its ID when CR sender is out of data frame but CR receiver still has data to send. On the other hand, CR receiver does not know whether it should stay and wait for CR sender data or back to control channel after QP when itself is out of data. Figure 5.2 illustrates how the NDBA works in normal cases (without switching the sender/receiver ID) and show the more dynamic part when there are new data frame come into receiver’s transmission queue. Figure 5.3 shows the switching sender/receiver ID cases of NDBA.

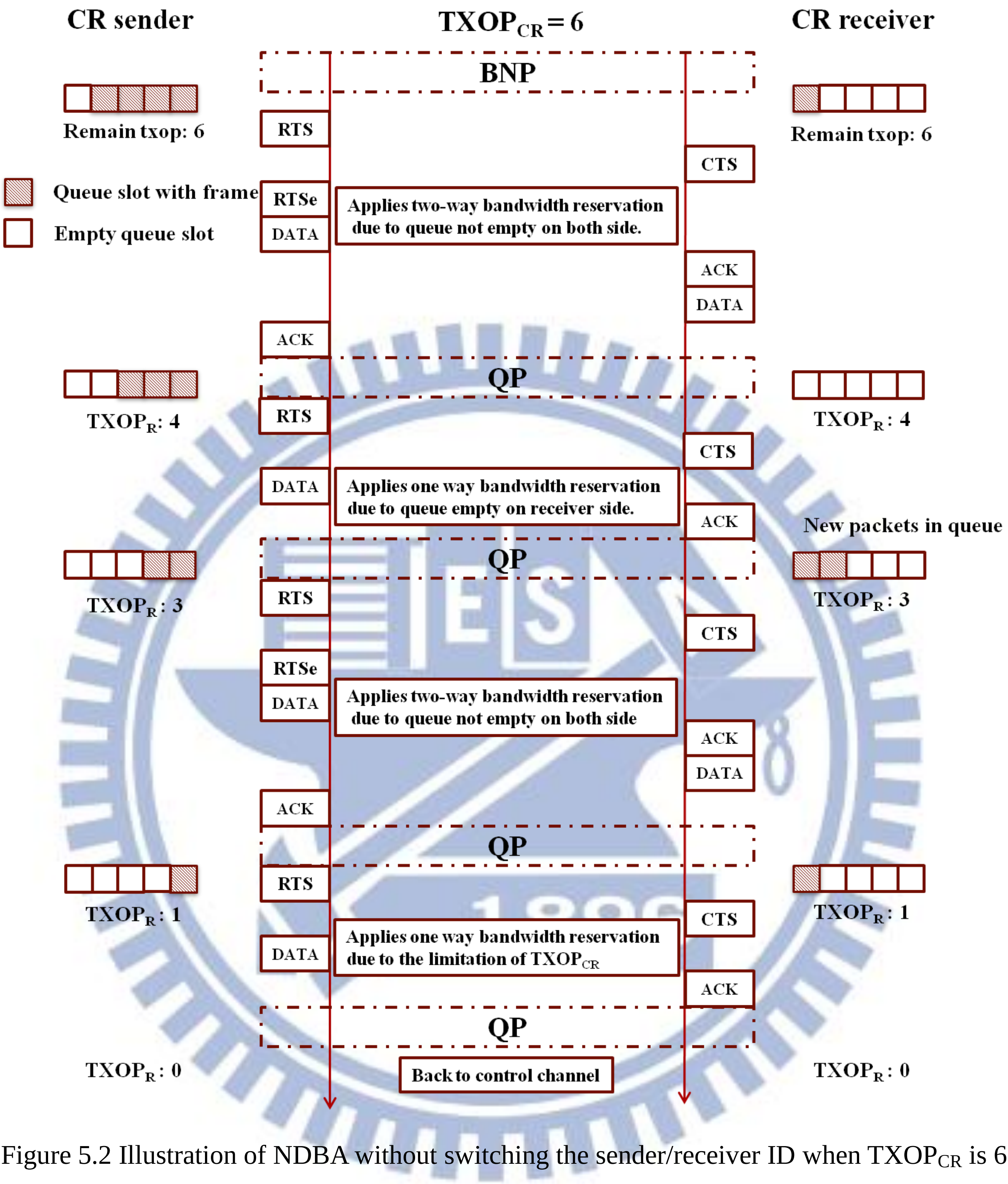


Figure 5.2 Illustration of NDBA without switching the sender/receiver ID when $TXOP_{CR}$ is 6

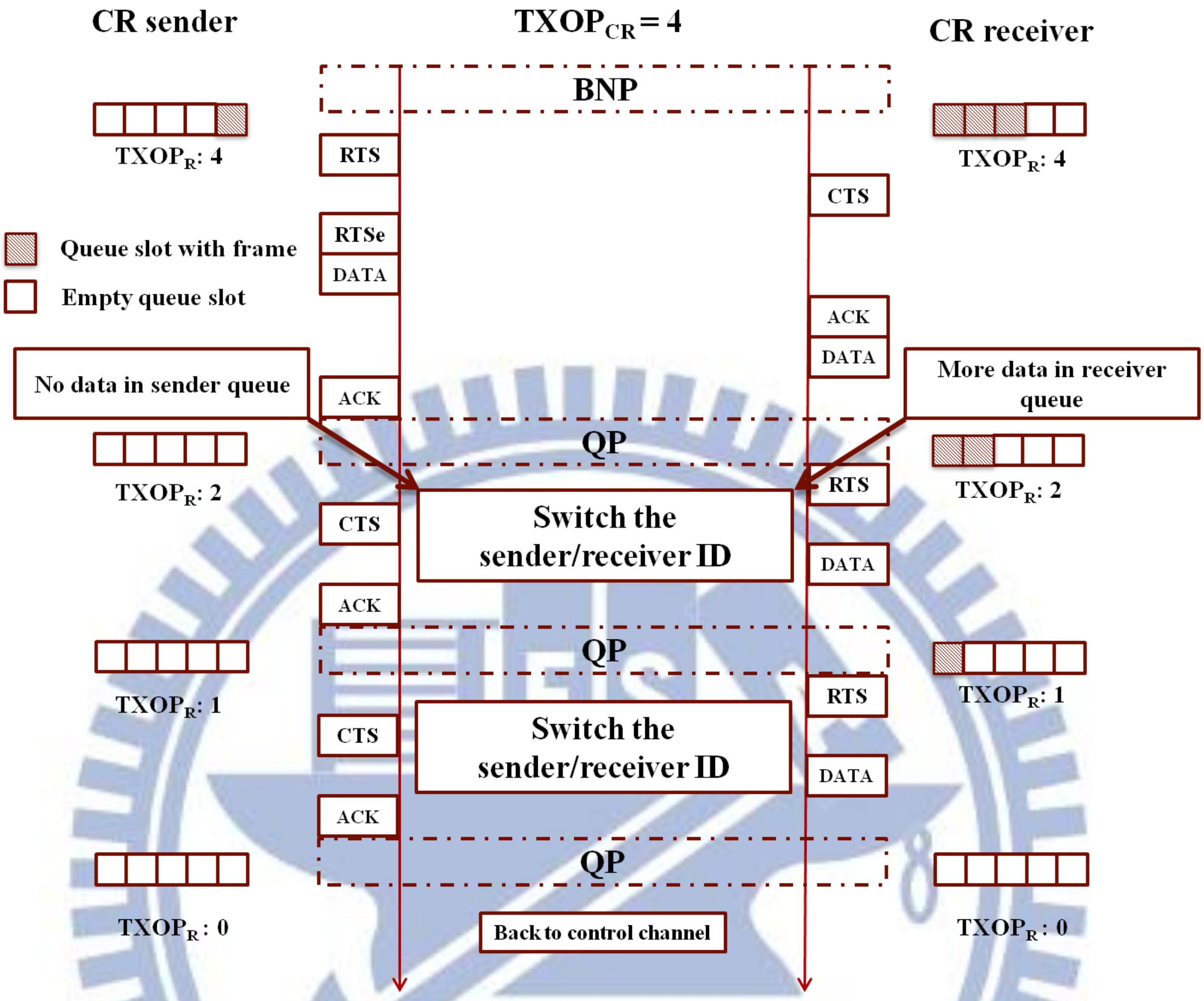


Figure 5.3 Switching Sender/Receiver ID Case of NDBA when $TXOP_{CR}$ is 4

5.3. New Bandwidth Negotiation Process

On the control channel, the CR MAC protocol uses IEEE 802.11 DCF which includes CSMA/CA and random backoff mechanism to prevent packets collision. In Ori-MAC, the control message RTS_{CR} and CTS_{CR} is modified from the RTS/CTS of IEEE 802.11 network. A CR sender starts BNP with transmits a RTS_{CR} to CR receiver. During the BNP, if the transmission of RTS_{CR} is failed due to the collision or unstable channel situation. The CR sender will notice that immediately because the CR sender won't receive CTS_{CR}. The CR sender will try to retransmit the RTS_{CR} again as soon as possible. However, in Uni-MAC and

ABi-MAC, the CR receiver will do the *fast channel sensing* procedure in order to collects the data channels states. That takes hundreds of microseconds before CR receiver replies $\mathrm{CTS}_{\mathrm{CR}}$ to CR sender. With this enhancement, if CR receiver does not receiver $\mathrm{RTS}_{\mathrm{CR}}$ correctly, the CR sender could only find out until the timer for waiting $\mathrm{CTS}_{\mathrm{CR}}$ expires. This could increase an overhead with hundreds of microsecond (see Figure 5.4 (b)). On the other hand, the $\mathrm{CTS}_{\mathrm{CR}}$ is sent by the CR receiver right after it finishes the *fast channel sensing*. This could cause the collision when another pair of CRU is also transmitting $\mathrm{RTS}_{\mathrm{CR}}$ or $\mathrm{CTS}_{\mathrm{CR}}$. The transmission failure of $\mathrm{CTS}_{\mathrm{CR}}$ to CR sender also causes the timeout of the timer for waiting $\mathrm{CTS}_{\mathrm{CR}}$ and the CR sender will try to retransmit $\mathrm{RTS}_{\mathrm{CR}}$ again (see Figure 5.4 (c)).

The above phenomenon happens more frequently while higher date rate is applied to CRUs. For example, it takes about 6 milliseconds to transmit a 1,500 bytes long frame with 2 Mbps data rate. The total transmission runs in one second is less due to a longer transmission time of a data frame. However, it takes only about 200 us to transmit a 1,500 bytes long frame with 54 Mbps data rates. The fast transmission of data frame increases the possible CR transmission rounds in one second that also increases the opportunity of collision in control channel. The number of CRU in the neighborhood also affects the probability the packet collision. The chance of collision become bigger when there is more CRUs nearby.

The original design of Uni-MAC and ABi-MAC doesn’t consider this problem. To sum up, the main problems of old BNP is (1) CR sender could only know the $\mathrm{RTS}_{\mathrm{CR}}$ transmission failure until the $\mathrm{CTS}_{\mathrm{CR}}$ waiting timer expires, (2) The CR receiver drop $\mathrm{CTS}_{\mathrm{CR}}$ directly when it notices control channel is busy after *fast channel sensing*. To avoid the problems above, we propose *New Bandwidth Negotiation Process* (NBNP) to improve the old BNP.

One can see in Figure 5.5 (a), the NBNP modifies the transmission of $\mathrm{RTS}_{\mathrm{CR}}$ and $\mathrm{CTS}_{\mathrm{CR}}$ to two way handshake process. We add an immediate ACK for both $\mathrm{RTS}_{\mathrm{CR}}$ and $\mathrm{CTS}_{\mathrm{CR}}$: $\mathrm{rRTS}_{\mathrm{CR}}$ replay the $\mathrm{RTS}_{\mathrm{CR}}$ and $\mathrm{rCTS}_{\mathrm{CR}}$ replay the $\mathrm{CTS}_{\mathrm{CR}}$. In this case, both CR sender and CR receiver can learn whether the transmission of RTS_{CR}/CTS_{CR} succeed or not right away. For example, CR sender can retransmit the RTS_{CR} if it does not receive the correlated CTS_{CR} right after the transmission of RTS_{CR} (see Figure 5.5 (b)). On the other hand, the CTS_{CR} now would try to retransmit itself if first transmission attempt after *fast channel sensing* fails (see Figure 5.5 (c)). We think the overhead of retransmission is much less then start over BNF again from CR sender.

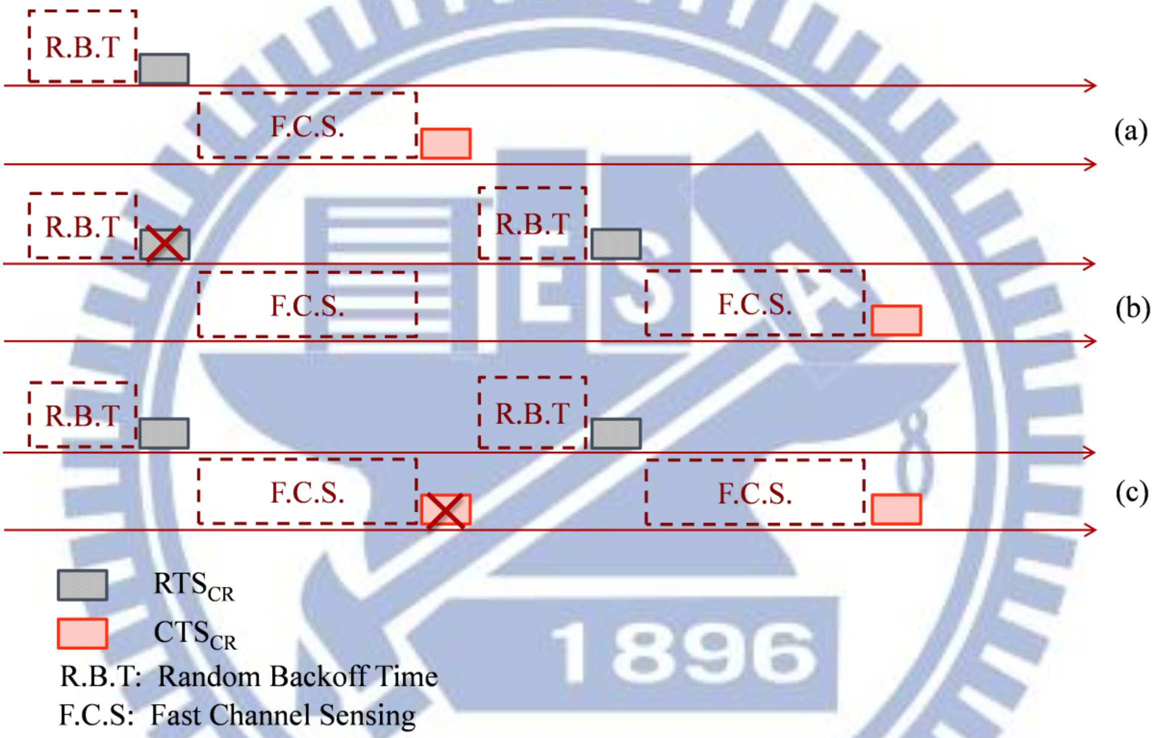


Figure 5.4 Illustration of Bandwidth Negotiation Process

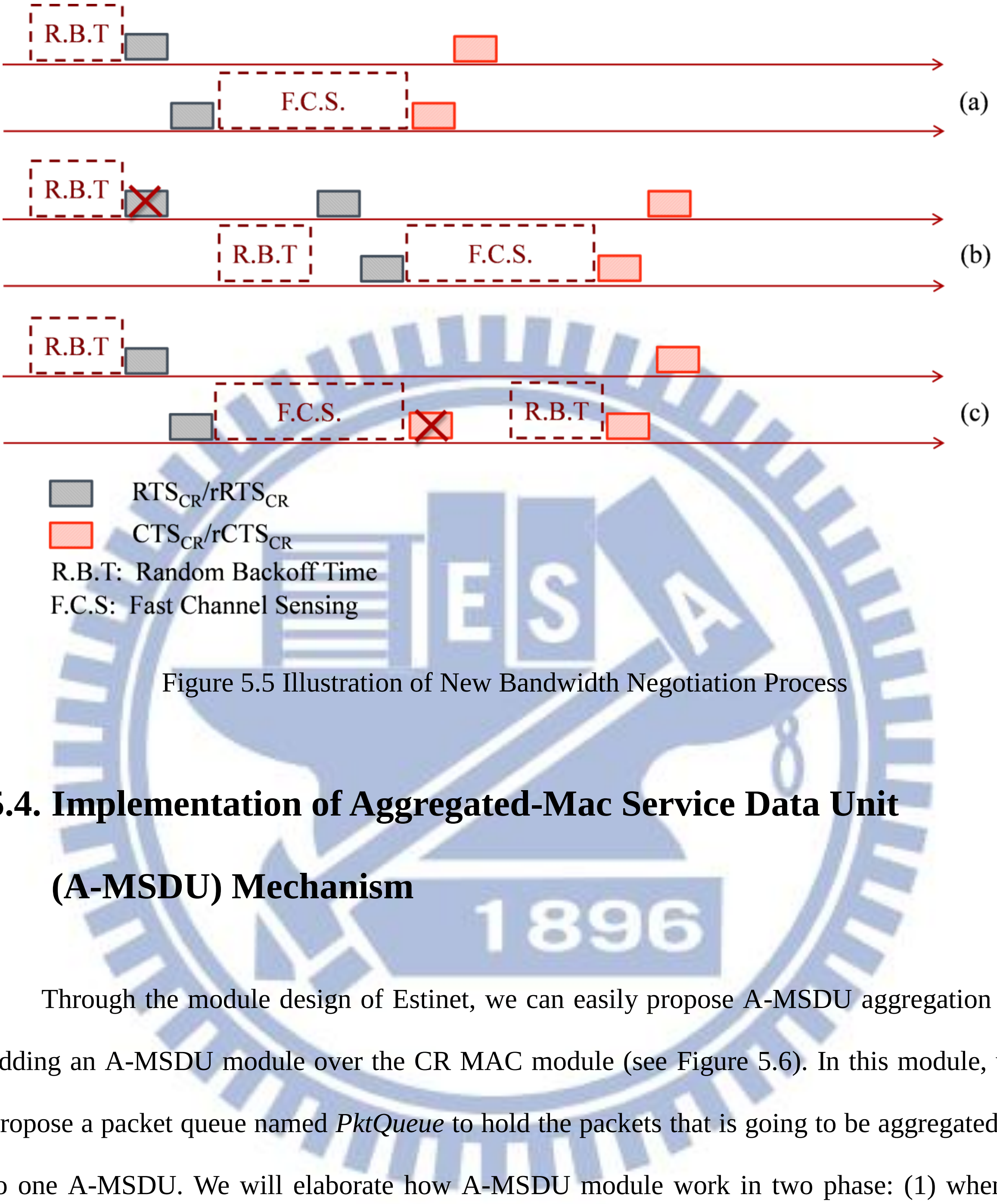


Figure 5.5 Illustration of New Bandwidth Negotiation Process

5.4. Implementation of Aggregated-Mac Service Data Unit (A-MSDU) Mechanism

Through the module design of Estinet, we can easily propose A-MSDU aggregation by adding an A-MSDU module over the CR MAC module (see Figure 5.6). In this module, we propose a packet queue named *PktQueue* to hold the packets that is going to be aggregated in to one A-MSDU. We will elaborate how A-MSDU module work in two phase: (1) when a packet is in the send direction and (2) a packet is in the receive direction.

In phase (1), the A-MSDU module monitors the MSDUs from upper layer (called incoming MSDU) and bypasses those broadcast packets directly. On the other hand, if the incoming MSDU is recognized as an IP packet, the A-MSDU module will try to do the aggregation process. The aggregation process first checks the length of *PktQueue*. If

PktQueue is empty, the incoming MSDU is queued in to *PktQueue*. If *PktQueue* is not empty, then the aggregation process compares the DA of the incoming MSDU and the DA of the head of *PktQueue*. If both DAs are not the same, the incoming MSDU is delivered to CR MAC module directly. Otherwise, the incoming MSDU will first check the available space of *PktQueue*. The available space of *PktQueue* is the different of maximum size of A-MSDU and the sum of MSDUs length in queue. In IEEE 802.11n [8], the maximum size of A-MSDU is ether 3389 bytes or 7935 bytes. We use 3389 bytes here in our setting. If the available size is larger than the length of incoming MSDU, the MSDU is queued in *PktQueue*. Otherwise, the module will aggregate the MSDUs in *PktQueue* because the total queue length reaches the maximum size of A-MSDU. The MSUDs are aggregated into an A-MSDU as mentioned in section 2.3.1. The aggregation will be trigger under another condition. When the first MSDU is queued in *PktQueue*, a 1 ms timer will be initiated. The timer will trigger the aggregation when it expires. This is the second condition to trigger the aggregation as mention in section 2.3.1. As a result, the A-MSDU is added with an *ampdu* packet info (packet info is the design used only in Estinet) and is delivered to CR MAC module.

In phase (2), the A-MSDU module monitors the MSDUs from CR MAC module at lower layer (also called incoming MSDU here). The A-MSDU module will recognize the A-MSDU due to the *amsdu* packet info. If the incoming MSDU is not coming with *amsdu* packet info, it is delivered to the upper layer directly. Otherwise, a de-aggregation process is triggered. Briefly, the de-aggregation process will de-aggregate A-MSDU into multiple MSDUs. After the de-aggregation process, the MSDUs are delivered to the upper layer.

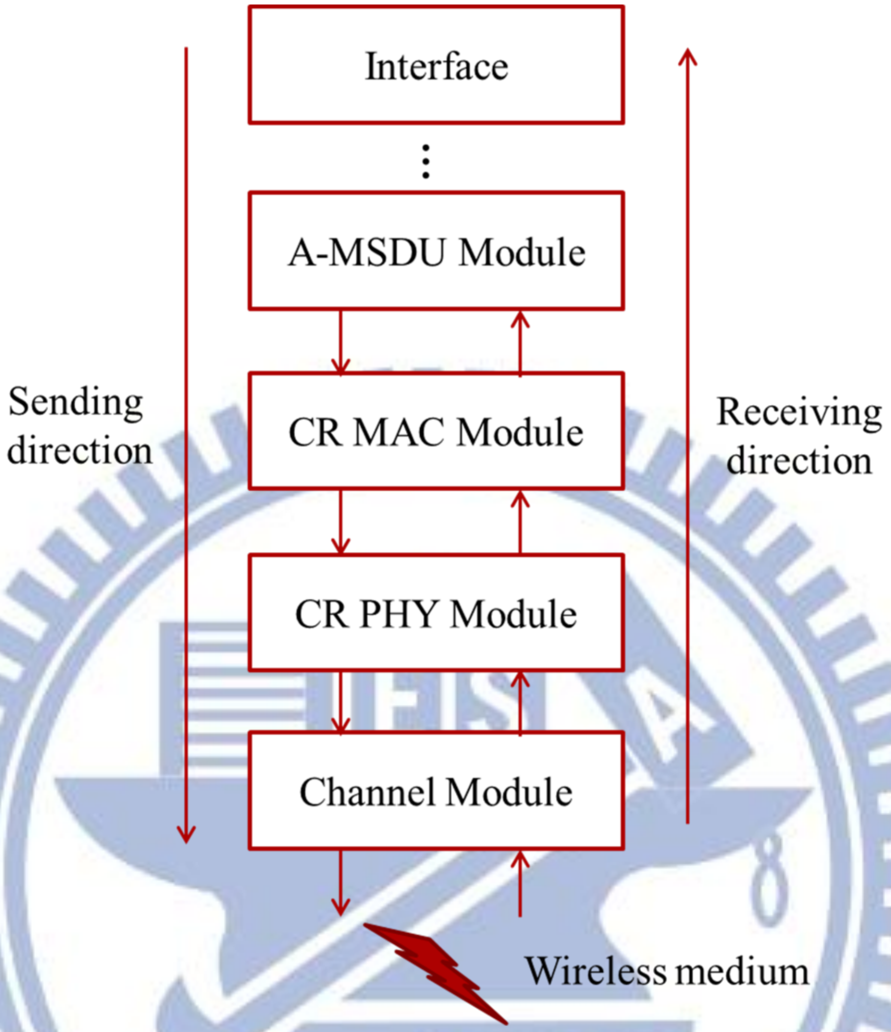


Figure 5.6 Illustration of CRU Protocol Modules Stack with A-MSDU module.

5.5. Implementation of Aggregated-Mac Protocol Data Unit (A-MPDU) Mechanism

In contrast with A-MSDU, we have to change the design of both Uni-MAC and ABi-MAC in MAC layer to accomplish the proposed A-MPDU implementation. First, we migrates the A-MPDU operation related function from Estinet 802.11n modules and modified them to fit our need. The main function of A-MPDU is *collect_MPDUs_into_frame_pack_* *for_AMPDU_operation* (called build AMPDU function for short in the following). This function aggregates the packet with same destination in transmission queue to an A-MPDU which is implemented by Estinet frame_pack data structure (see Figure 5.7). An aggregated MPDU is first concatenated by a MPDU delimiter in front of it, set dh_qos.ack_policy header field to 3, and chain in to frame_pack and each MPDU is add with an extra *AMPDU pkt_info* for recognition. The *AMPDU pkt_info* is used only in Estinet simulation engine. If a frame with *AMPDU pkt_info* is recognized in physical layer of CRU, the simulation engine will reduce the PHY header and PLCP header transmission time from the total frame transmission time. If a MPDU with *AMPDU pkt_info* is received in CRU's MAC layer, the additional decapsulation of MPDU delimiter is perform before transmitting the MPDU to upper layer. Through the mechanisms mentioned above, Uni-MAC and ABi-MAC can now support A-MPDU frame aggregation.

```
struct frame_item {
          struct ac_entity          *ac_ent_;
          Event_                     *ep_;
          u_int32_t                  channel;
          double                     data_rate;
          double                     tx_power;
          struct frame_item         *pre_;
          struct frame_item         *nxt_;
};
struct frame_pack {
          enum FramePackAction      action;
          u_int32_t                  num_frame;
          u_int32_t                  total_len_of_all_frames_in_bytes;
          u_char                     dst_addr[ETHER_ADDR_LEN];
          u_int64_t                  lifetime_in_tick;
          u_int16_t                  nav_duration_in_us;
          u_int16_t
nav_duration_in_us_to_minus_for_block_head_frame;
          u_int8_t                   tid;
          u_int16_t                  start_seq;
          struct frame_item         *search_position_;
          struct frame_item         *head_;
          struct frame_item         *tail_;
};
```

Figure 5.7 The Structure of frame_item And frame_pack.

In Uni-MAC, the number of MPDU aggregated is according to $TXOP_{CR}$ setting. In ABi-MAC, the number of aggregated frames is decided by the new dynamic allocation mechanism as mention in section 5.2. However, the situation is more complex due to the design of A-MPDU agregation that CRUs can transmit mutiple data frames in one data transmission turn of ABi-MAC. So we define the aggregated A-MPDU size can not bigger than half of $TXOP_{CR}$ value. This is for the fairness to CR receiver so it can send at most the same data frame as CR sender. If the situation of asymetric flow as shown in Figure 5.8 happened, CR sender can take the remaining TXOP$_{CR}$ after QP.

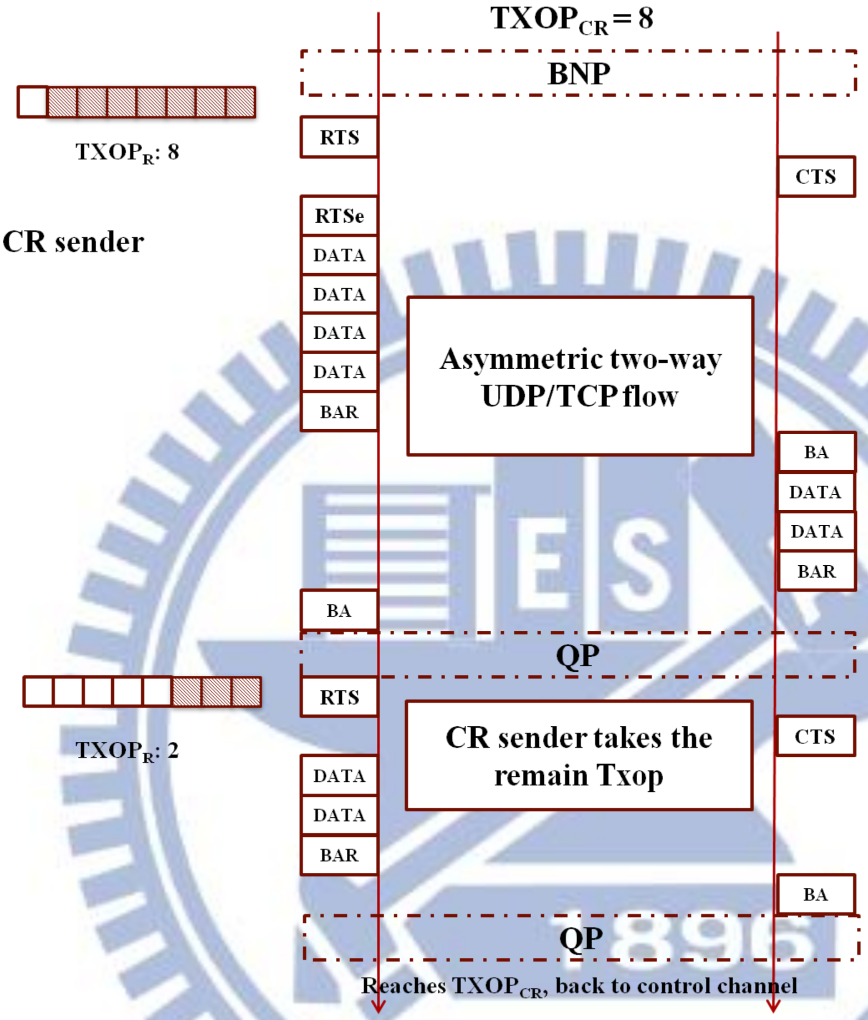


Figure 5.8 Illustration of A-MPDU frame aggregation and NDBA with asymetric flow.

Figure 5.9 illustrate the operation of Uni-MAC with both A-MPDU and block ACK mechanisms are applied and the value of TXOP$_{CR}$ is 6. Figure 5.10 illustrate the operation of ABi-MAC with both A-MPDU and block ACK mechanisms when running a symetric two-way traffic flow on both CR sender and CR receiver and the value of TXOP$_{CR}$ is 6.

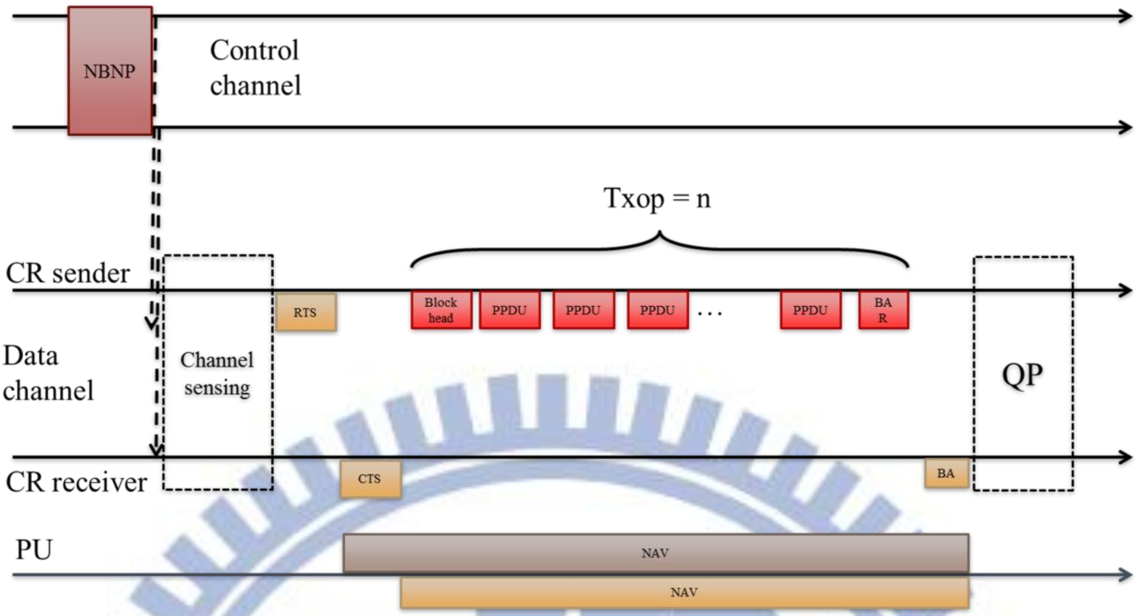


Figure 5.9 Illustration of A-MPDU And Block ACK Operation Over Uni-MAC (TXOP$_{CR}$ = n)

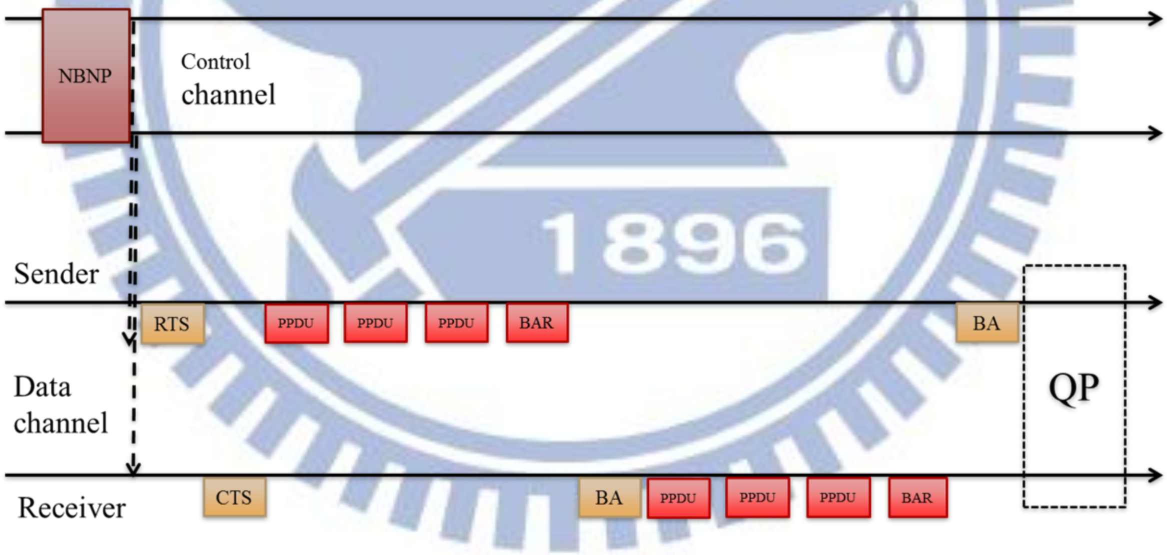


Figure 5.10 Illustration of A-MPDU and Block ACK operation over ABi-MAC with symmetric traffic flow (TXOP$_{CR}$ = 6)

5.6. Implementation of Block ACK Operation

To achieve the purpose of block ACK operation, the following are the most important function:

- store_data_under_Block_Ack_operation (store function)
- build_ba_and_release_data_based_on_received_bar (BA function)
- remove_acked_data_within_block_based_on_received_ba (receive BA function)

First, the store function is called when a CRU receives a data frame with dh_qos.ack_policy value is 3. This means the data frame is belong to a set of MPDUs operating block ACK operation. The function then stores the MPUD into a list and keeps waiting for MPDU until received a BAR.

While receiving a BAR, the receiver will call BA function to build the BA. In BA function, it go through the list that keeps the received MPDUs and calculate the MPUD's *offset* from its sequence number minus the sequence number of first received MPDU. Then the function maps the offset of each MPDU into an 8 by 8 matrix. After creating BA, the MAC module starts to transmit the MPDU in the list to the upper layer. At last, the sender receives the BA and calls receive BA function to remove the MPDU that is acknowledged in BA.

```
struct block_ack_data_storage {
            u_int8_t                      tid;
            u_int32_t                     num_src;
            struct block_ack_src        *search_position_;
            struct block_ack_src        *head_;
            struct block_ack_src        *tail_;
};
```

Figure 5.11 The Structure of frame_item and frame_pack

Chapter 6 Performance evaluation

In this chapter, we use the Estinet simulator to evaluate the performance of FA-MAC with different simulation cases. One can see the performance enhancement when different frame aggregation mechanism is applied to Uni-MAC and ABi-MAC. The influence to PUs from the applied frame aggregation mechanism is another important issue. We will first introduce our simulation topologies and parameter settings. Next, we will show the improvement of maximum throughput while three different frame aggregation mechanism is applied: A-MSDU, A-MPDU, and two-level aggregation (TLA). Last, we show the influence to PUs' protection while different frame aggregation mechanism is applied and end this chapter with the summary. In order to make the reading more clearly, we will use the abbreviation listed in Table 6.1 below.

Table 6.1 The abbreviation of different CR MACs and Frame Aggregation Mechanisms

Original Uni-MAC	Uni-MAC
Uni-MAC applied with A-MSDU only	Uni-AMSDU
Uni-MAC applied with A-MPDU only	Uni-AMPDU
Uni-MAC applied with Two-Level Aggregation	Uni-TLA
Original ABi -MAC	ABi -MAC
ABi -MAC applied with A-MSDU only	ABi -AMSDU
ABi -MAC applied with A-MPDU only	ABi -AMPDU
ABi -MAC applied with Two-Level Aggregation	ABi -TLA

6.1. Simulation Settings

We use two different simulation cases to evaluate the performance of FA-MAC: (1) single pair of CRUs and (2) 5 pairs of PUs and 5 pairs of CRUs. The common parameter setting of simulation is addressed in Table 6.2. The network topologies and applied traffic flows are introduced case by case in the following.

Table 6.2 The parameter settings of CR MAC protocol

Parameter Name	**Value**
Number of Data Channel	5
Channel Bandwidth	20 MHz
Data Rate	54 Mbps
Channel Sensing Period	200 us
DIFS	41 us
SIFS	16 us
Duration of QP	200 us
Txop	1-5 (Uni-MAC) / 2-10 (ABi-AMC)
Applied Frame Aggregation Mechanism	A-MSDU, A-MPDU, TLA

Simulation Case 1:

This case is for the measurement of the CR MAC maximum throughput without other PUs or CRUs. The topology only contains one pair of CRU as shown in Figure 6.1. The CRU pair is in their transmission range and the traffic flows applied to each CRU are shown in Table 6.3. In this case, we can find out whether ABi-MAC still outperforms than Uni-MAC after frame aggregation mechanisms are applied.

Figure 6.1 Simulation Case 1 for Maximum Throughput Evaluation

Table 6.3 Traffic flows Setting of Case 1

Type	**CR sender**	**CR receiver**
Uni-MAC UDP Maximum Throughput	Greedy UDP flow with different packet size (100, 200, 500, 1000, 1450 bytes)	none
Uni-MAC TCP Maximum Throughput	Greedy TCP flow	none
ABi-MAC UDP Maximum Throughput	Greedy UDP flow with different packet size (100, 200, 500, 1000, 1450 bytes)	Greedy UDP flow with different packet size (100, 200, 500, 1000, 1450 bytes)
ABi-MAC TCP Maximum Throughput	Greedy TCP flow	none

Simulation Case 2:

This case has a more complicated topology in order to observe the influence from CRUs to PUs (see Figure 6.2). The reason of using 5 pairs of PU is because there are 5 data channels. Each pair of PUs operate on an individual data channel so PUs will not interfere each other. In order to observe the influence from CRUs , we use 5 pairs of CRUs that each data channel may has the same chance to be access by CRUs in average.

The PU pair will generate a one-way UDP flow with the load of 30%, 50%, and 80% of maximum throughput when 54 Mbps data rate is applied. The different loads of PU represent different busy degree of a data channel. The different loads packet flow is generated by

Estinet *stg* tool with –i option. The stg with –i option will read a configuration file and generate the network flow accordingly. We can use the configuration file to generate various network traffic that we need. In this case, we use *stg* to generation the traffic flow for PU which the packet interval is a Poisson distribution and the throughput is 30%/50%/80% of PU's maximum throughput when 54 Mbps data rate is applied. The configuration detail is shown in Table 6.4 and the traffic flow applied to CRUs is shown in Table 6.4. We applied Greedy TCP/UDP flow for CRU for estimating the performance of CRU pair with the existence of PUs and other CRU and the influence to PUs.

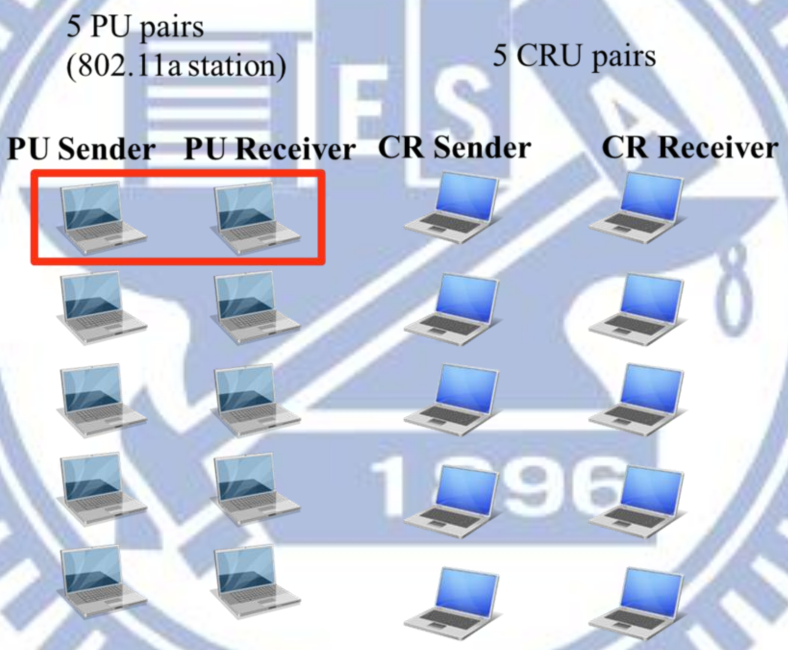


Figure 6.2 Simulation case 2 for evaluating the influence to PUs

Table 6.4 PUs traffic flow and the configuration command

Loads	Configuration command
30% Load	on: time: 12 exponential 0.0015 0.001 0.002 length: const 1450
50% Load	on: time: 12 exponential 0.0008 0.0006 0.001 length: const 1450
80% Load	on: time: 12 exponential 0.0005 0.0003 0.0008 length: const 1450

On: time:12: this flow should run 12 seconds.

exponential XXX YYY ZZZ: means the interval of packet is an exponential distribution with MEAN=XXX, MAX=YYY, MIN=ZZZ (the unit is micro second).

length: const 1450: the generated packet size is constant 1,450 byte.

Table 6.5 The network flow applied to CRUs

CR MAC and Traffic flow	**CR sender**	**CR receiver**
ABi-MAC UDP Flow	Greedy UDP flow	Greedy UDP flow
ABi-MAC TCP Flow	Greedy TCP flow	None

Bit Error Rate (BER):

In the above two cases, we also add different bit error rates to each FA-MAC to evaluate the robustness with the influence of different wireless spectra condition. The BER here is defined as the probability of bit error happens after the receiver finishes the channel coding. Through this evaluation, we can know the reliability of different FA-MACs.

6.2. The Improvement of Maximum Throughput of FA-MAC

We can see in Figure 6.3 that all three kinds of frame aggregation mechanism do improve the maximum throughput of Uni-MAC. First, we look at the maximum throughput improvement of Uni-AMSDU. The throughput increases about 70% (see Table 6.6) with each TXOP$_{CR}$ value. The reason is that with the A-MSDU aggregation, CR sender can transmit at most 2 UDP packet aggregated in one A-MSDU at a time (the A-MSDU maximum size is set to 3839 that can contains up to two 1,450 byte long UDP packets). Second, the result of Uni-AMPDU is different from Uni-AMSDU. When TXOP$_{CR}$ value is one, Uni-AMPDU works same as Uni-MAC but with little additional overhead from A-MPDU operation (BAR and BA). That’s why the throughput of UNI-AMPDU is 1% less the Uni-MAC. The throughput improvement becomes better when $\mathrm{TXOP}_{\mathrm{CR}}$ increases because Uni-AMPDU can transmit at most $\mathrm{TXOP}_{\mathrm{CR}}$ data frames in a data transmission round. The bigger $\mathrm{TXOP}_{\mathrm{CR}}$ value reduces more transmission overhead. Last, one can see from the curve of the improvement of Uni-TLA, which combines the improvement from the above two frame aggregation mechanisms.

The result of TCP flow is shown in Figure 6.4 and Table 6.7. We can see the curves grow almost the same trend as the result of UDP in Figure 6.3. The difference is that the improvement degree of Uni-AMSDU and Uni-TLA become less when $\mathrm{TXOP}_{\mathrm{CR}}$ increases. This is because TCP can treat as a two-way asymmetric traffic flow and some of the transmission opportunity is taken by the small TCP ACK packets.

Table 6.6 The improvement of Maximum UDP Throughput when different Frame Aggregation Mechanism is applied. (in percentage)

	$TXOP_{CR}$ = 1	$TXOP_{CR}$ = 2	$TXOP_{CR}$ = 3	$TXOP_{CR}$ = 4	$TXOP_{CR}$ = 5
UNI-AMSDU	78.70%	70.71%	66.46%	63.78%	61.94%
UNI-AMPDU	-1.08%	14.11%	32.32%	47.52%	60.39%
UNI-TLA	76.81%	91.93%	109.48%	123.26%	134.09%

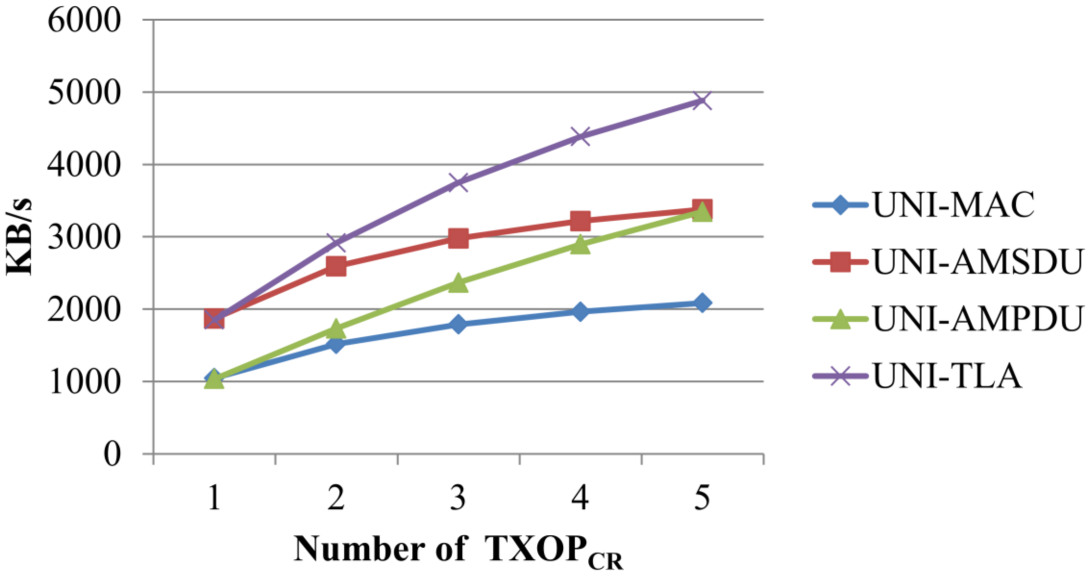


Figure 6.3 Maximum UDP Throughput of Uni-MAC with different Frame Aggregation Mechanism (with UDP payload 1450 bytes)

Table 6.7 The improvement of Maximum TCP Throughput when different Frame Aggregation Mechanism is applied (in percentage)

	$TXOP_{CR}$ = 1	$TXOP_{CR}$ = 2	$TXOP_{CR}$ = 3	$TXOP_{CR}$ = 4	$TXOP_{CR}$ = 5
UNI-AMSDU	93.91%	72.43%	57.34%	49.75%	44.60%
UNI-AMPDU	-1.30%	26.98%	32.16%	36.13%	39.31%
UNI-TLA	92.33%	83.31%	75.91%	72.26%	69.56%

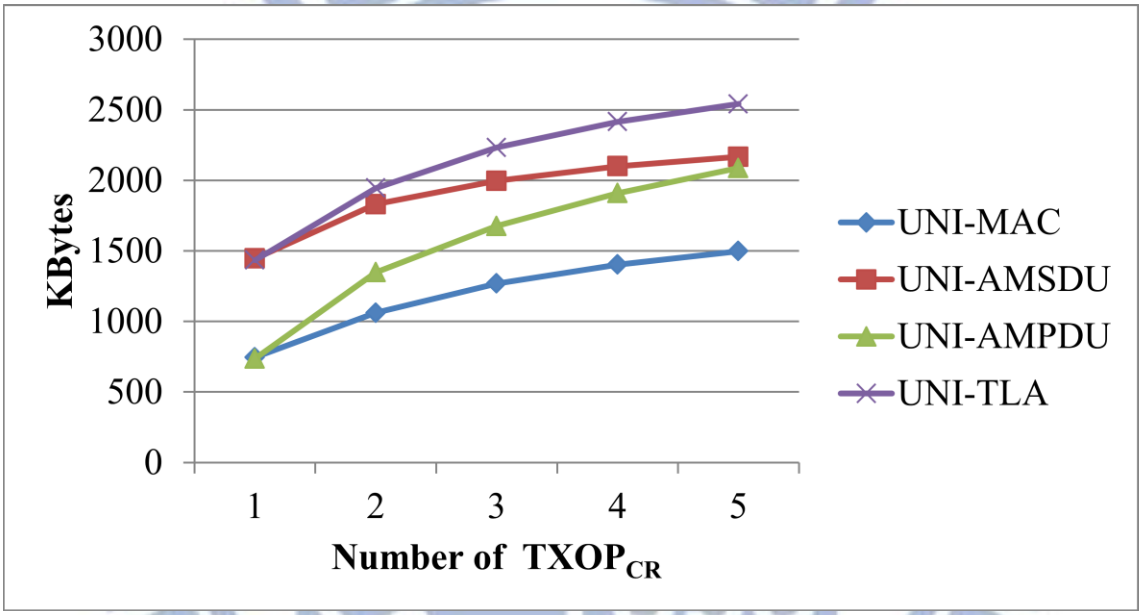


Figure 6.4 Maximum TCP Throughput of Uni-MAC with different Frame Aggregation Mechanism

- **The maximum throughput improvement of ABi-MAC with different frame aggregation mechanism is applied:**

To evaluate the maximum throughput of UDP flow of ABi-MAC, we put a greedy UDP traffic flow (with 1450 byte UDP payload) on both side of CRU pair due to the bi-direction bandwidth reservation design. So the maximum throughput is now defined as the sum of received throughput on both side CRU pair in cases 1. Due to the design of A-MPDU over

ABi-MAC in section 5.5 we can only use the $TXOP_{CR}$ value which is multiplies of two. One should note this when comparing the result of Uni-MAC with ABi-MAC.

In Figure 6.5 and Table 6.8, the improvement of A-MSDU is still obvious and the improvement of A-MPDU is also increase with the $TXOP_{CR}$ value. The improvement of ABi-TLA is the sum of the improvement of ABi-AMSDU and Bi-AMPDU.

Table 6.8 The improvement of Maximum UDP Throughput when different Frame Aggregation Mechanism is applied (in percentage)

	$TXOP_{CR} = 2$	$TXOP_{CR} = 4$	$TXOP_{CR} = 6$	$TXOP_{CR} = 8$	$TXOP_{CR} = 10$
ABi-AMSDU	62.694%	56.302%	52.912%	50.761%	49.014%
ABi-AMPDU	-23.364%	-1.601%	14.398%	28.116%	38.626%
ABi-TLG	30.250%	54.447%	69.239%	80.636%	88.279%

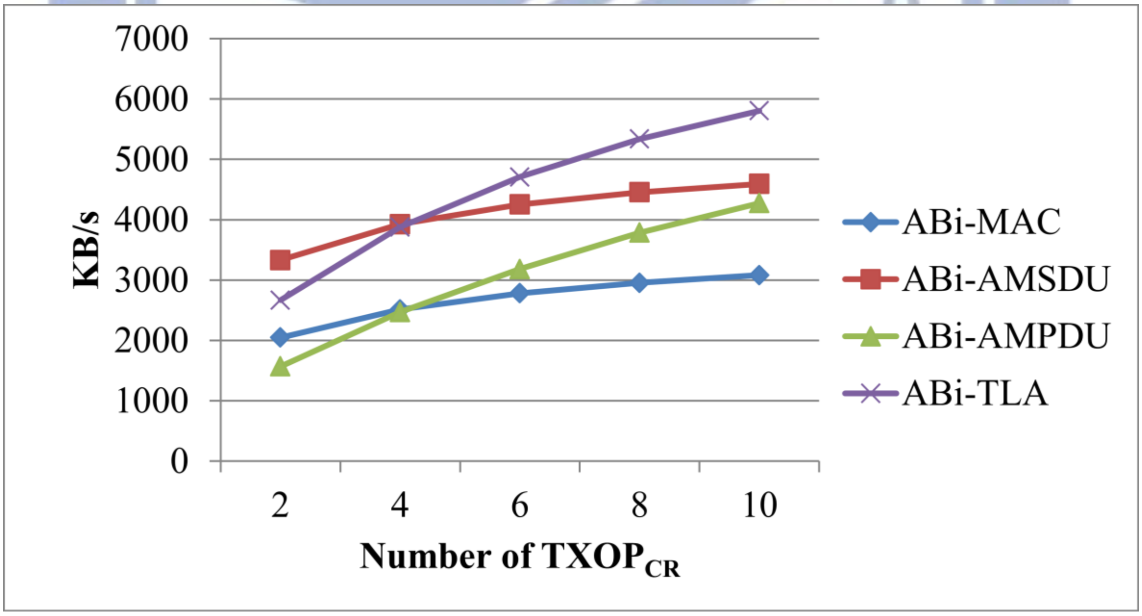


Figure 6.5 Maximum UDP Throughput of ABi-MAC with different Frame Aggregation Mechanism (with UDP payload 1450 bytes)

Figure 6.6 and Table 6.9 shows the maximum throughput of TCP flow. One can see the TCP throughput is better than the TCP throughput of UNI-MAC (in Figure 6.4) with same TXOP_{CR}. This is because of the two-way bandwidth reservation design of ABi-MAC. The design not only reduces the RTT of TCP flow but also reduces the overhead of QP.

The improvement of ABi-AMSDU becomes bigger (from 28% ~ 74%) with the bigger TXOP_{CR} (see Table 6.9) which is different from the result of UNI-MAC (from 93% ~ 45% in Table 6.7). This is because with A-MSDU frame aggregation, many TCP ACKs can aggregates into a single A-MSDU and save the overhead. However, according to the design of Uni-MAC, after TCP sender transmits the TCP packets, the TCP receiver needs an additional CR transmission round to reply the TCP ACK. At this moment, the TCP receiver may note use all the TXOP_{CR} due to the A-MSDU aggregation which aggregates several TCP ACK into an A-MPDU. On the other hand, the TCP receiver can reply the aggregated ACKs in between the transmission of TCP sender's packet through the design of ABi-MAC that utilize the transmission opportunity in a more efficient way. The same phenomenon happens when A-MPDU is applied, that's why the improvement of ABi-AMPDU has a bigger improvement with bigger TXOP_{CR} (improves 52% when TXOP_{CR} is 5 and the improvement of Uni-AMPDU is 39% when TXOP_{CR} is 5). Due to the above reasons, the improvement of ABi-TLA is bigger when bigger TXOP_{CR} is applied.

Table 6.9 The Improvement of Maximum TCP Throughput when different Frame Aggregation Mechanism is applied. (in percentage)

	$TXOP_{CR}$ = 2	$TXOP_{CR}$ = 4	$TXOP_{CR}$ = 6	$TXOP_{CR}$ = 8	$TXOP_{CR}$ = 10
ABi-AMSDU	28.70%	73.09%	77.54%	75.19%	74.21%
ABi-AMPDU	-22.78%	9.42%	26.98%	40.56%	52.73%
ABi-TLG	39.36%	93.28%	123.35%	121.58%	131.08%

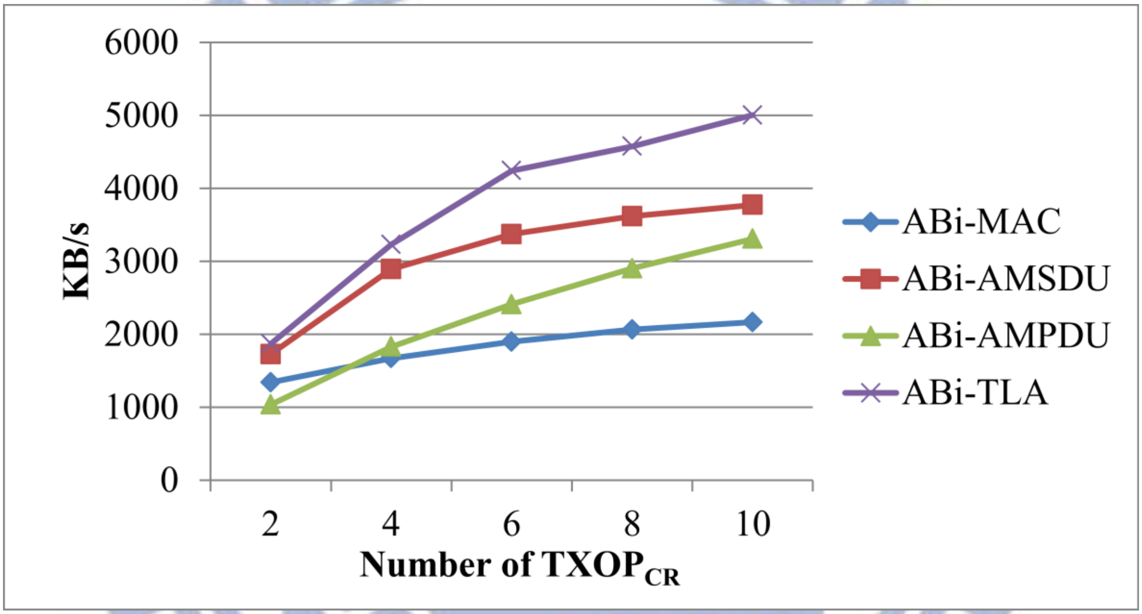


Figure 6.6 Maximum TCP Throughput of ABi-MAC with different Frame Aggregation Mechanism

In the following section, we add different bit error rate in the simulation to evaluate the performance of FA-MAC's affected by different bit error rate.

- **The maximum UDP throughput of ABi-MAC with different bit error rates:**

Figure 6.7 shows the application layer throughput of UDP throughput of ABi -MAC with different bit error rate. First, one can see that the performance of FA-MAC degrades slightly when applied BER is 10^{-6}. Second, the throughput of ABi-AMSDU and ABi-TLA degrade more than the throughput of ABi-MAC and ABi-AMPDU when BER becomes 10^{-5}. This is because the size of single of frame of ABi-AMSDU and ABi-TLA is bigger than the frame size of ABi-MAC and ABi-AMPDU due to the A-MSDU aggregation mechanism. The bigger MPDU has more probability to encounter the bit error. Once the bit error occurs, the CRU sender won't get the ACK reply and should hop back to the control channel that hardly decreases the efficiency. On the other hand, the ABi-AMPDU can transmit all the collected MPDUs with A-MPDU mechanism and block ACK even if there is an error occurs in one of the frames. That's why the degradation of the throughput of ABi-AMPDU is slightly than the throughput of ABi-AMSDU. Last, since the single data frame size is bigger than 10,000 bits (1,500 bytes * 8 = 12,000 bits), all FA-MACs cannot operate when high BER likes 10^{-4} is applied.

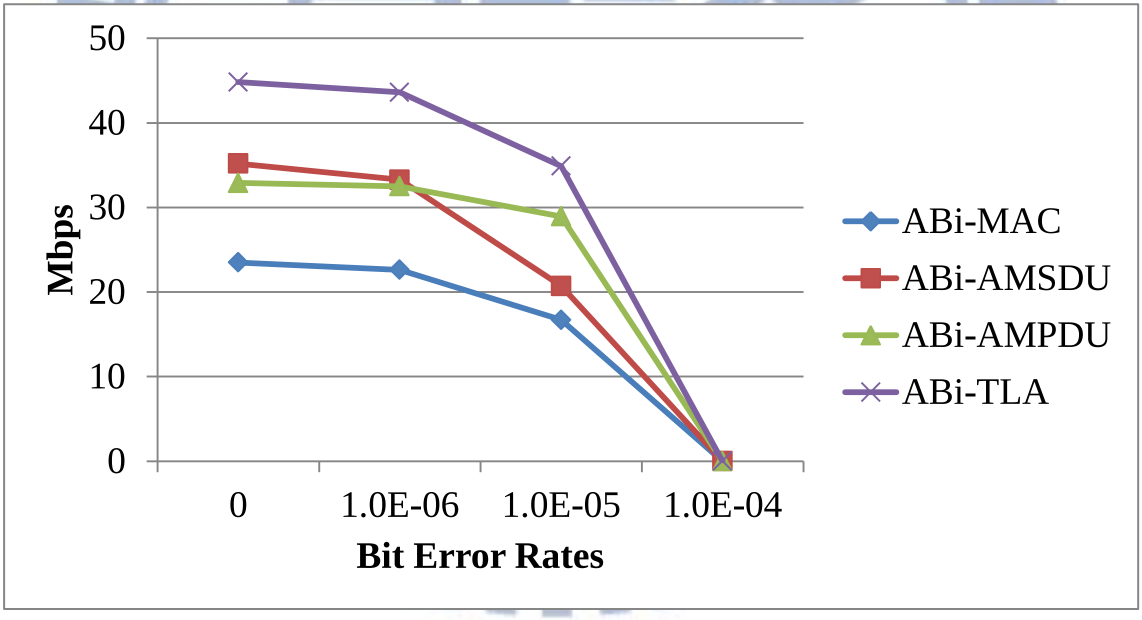


Figure 6.7 Maximum Application Layer UDP Throughput of ABi -MAC with different Bit Error Rates (with UDP payload 1450 bytes, $TXOP_{CR}$ = 10)

- **The maximum TCP throughput of ABi-MAC with different bit error rates:**

Figure 6.8 shows the application layer throughput of TCP throughput of ABi-MAC with different bit error rate. First, one can see that the performance of FA-MAC degrades slightly when applied BER is 10^{-6} just likes the result of UDP throughput above. Second, the throughput degradation of ABi-AMSDU is bigger than ABi-AMPDU due to the big frame aggregated by the A-MSDU aggregation. The reason is same as the UDP case as mention in the previous section. Finally, all FA-MACs cannot operate when 10^{-4} BER is applied since TCP payload is 1448 bytes long that cannot bear such the high data rate.

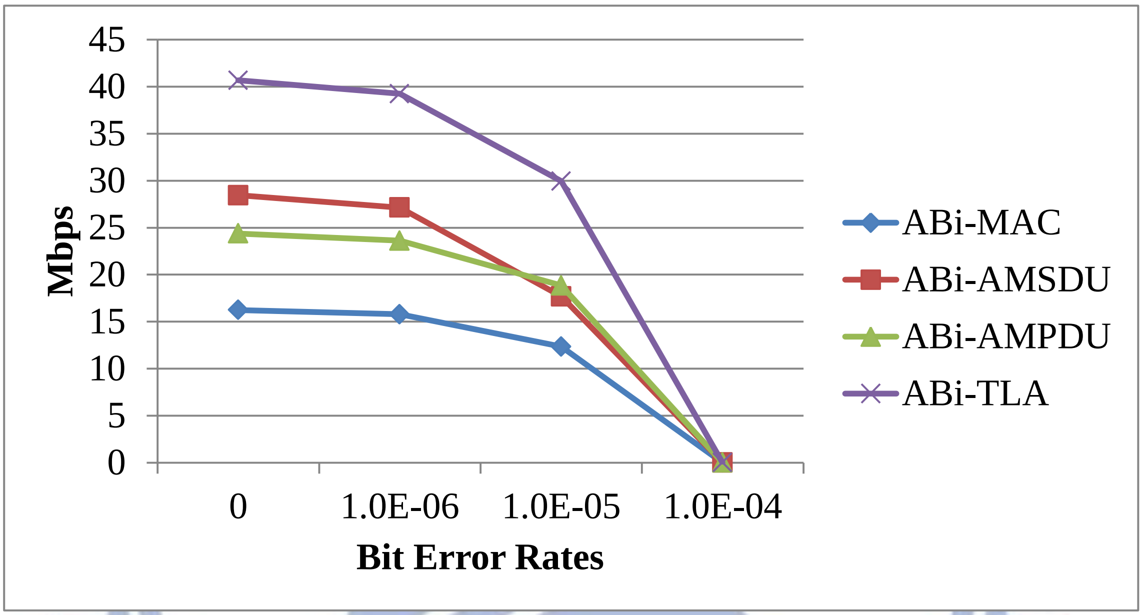


Figure 6.8 Maximum Application Layer TCP Throughput of ABi-MAC with different Bit Error Rates ($TXOP_{CR}$ = 10)

6.3. The Effect on Primary Users' Throughput and Packet Transmission Delay from FA-MAC

In this chapter, we want to see the influence of CRU when it tries to use the channel with other PUs in the neighborhood. We only evaluate ABi-MAC with different frame aggregation mechanism in this case. Through the maximum throughput measurement result of both Uni-MAC and ABi-MAC, we can find out ABi-MAC is still better than Uni-MAC in all respect. Thus, we only use ABi-MAC for further testing.

We apply three different channel busy degrees to evaluate the influence on PUs as mentioned in sec 6.2. Each PU pair applies 30%, 50%, and 80% of maximum UDP load (with 1,450 UDP payload) on their own channel to represent light, medium, and heavy channel busy degree. With different degree, we can learn the performance of FA-MAC (ABi-MAC applied with different frame aggregation mechanisms in this chapter) more completely. For each load degree, we will show the result of ABi-MAC with different frame aggregation mechanism and different traffic flow (TCP/UDP). The result is discussed in the following.

- **ABi-MAC with 30% PU load**

In Table 6.10, we can see that the throughput of PU still remain the same at about 1,050 KB/s with the influence of CRUs. This result shows that the protection to PUs of ABi-MAC works properly with different frame aggregation mechanism is applied. On the other hand, the throughput of CRU pair still benefit from frame aggregation mechanism as shown in Figure 6.9. First, the improvement of A-MPDU is still obvious when comparing the curves of ABi-MAC and ABi-AMSDU. However, the curves of ABi-MAC and ABi-AMSDU stop growing when TXOP_{CR} reaches four. This is because of the channel vacate mechanism of ABi-MAC. In this case, the average packet generation interval of 30% load is about 1.5 ms (see the setting in Table 6.4). In average, CRU would hear the channel claiming request from PU after transmits two frames in one CR transmission round. This is the reason why the throughput stops growing while TXOP_{CR} is bigger than four.

Second, the improvement of ABi -AMPDU aggregation grows bigger while the TXOP_{CR} is bigger. Due to A-MPDU aggregation and block ACK mechanism, ABi-AMPDU and ABi-TLA can both transmit at most TXOP_{CR} frames in one CR transmission round even when TXOP_{CR} is larger than three. However, this will increase the packet delay of PUs packet when PUs wants to claim the channel but the channel is reserved. On can see average packet delay time in Figure 6.10. Uni-TLA causes the most delay because it would occupied the channel for the longest time with both frame aggregation mechanisms are applied.

The situation is pretty much the same when CRU is applying TCP flow. First, the throughput of PU still remains the same at about 1,050 KB/s with the influence of CRUs in Table 6.11. Second, the curve of ABi-MAC and ABi-AMSDU stop growing at the same TXOP_{CR} value. Last, ABi-AMPDU and ABi-TLA cause the biggest PU’s packet delay time.

Table 6.10 The PU's Throughput of 30% Load with the effect of CRUs (KB/s)

	$TXOP_{CR} = 2$	$TXOP_{CR} = 4$	$TXOP_{CR} = 6$	$TXOP_{CR} = 8$	$TXOP_{CR} = 10$
ABi-MAC	1050.644	1050.461	1050.492	1050.431	1050.644
ABi-AMSDU	1050.461	1050.735	1050.735	1050.431	1050.37
ABi-AMPDU	1050.339	1050.613	1050.613	1050.613	1050.339
ABi-TLA	1050.766	1050.431	1050.37	1050.492	1050.248

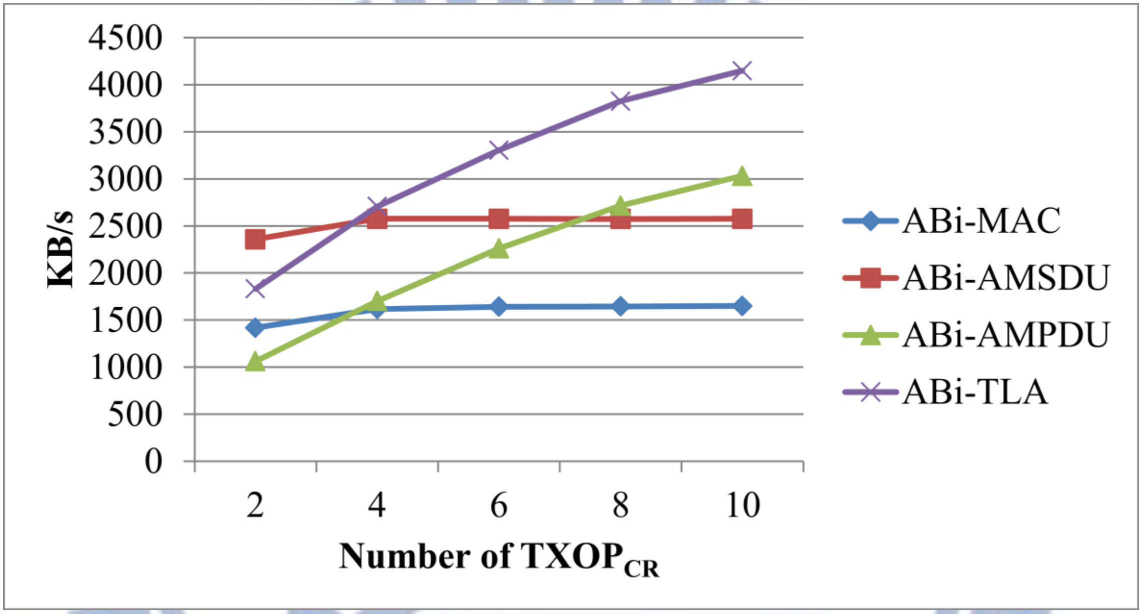


Figure 6.9 UDP Throughput of CRU with 30% PU Load

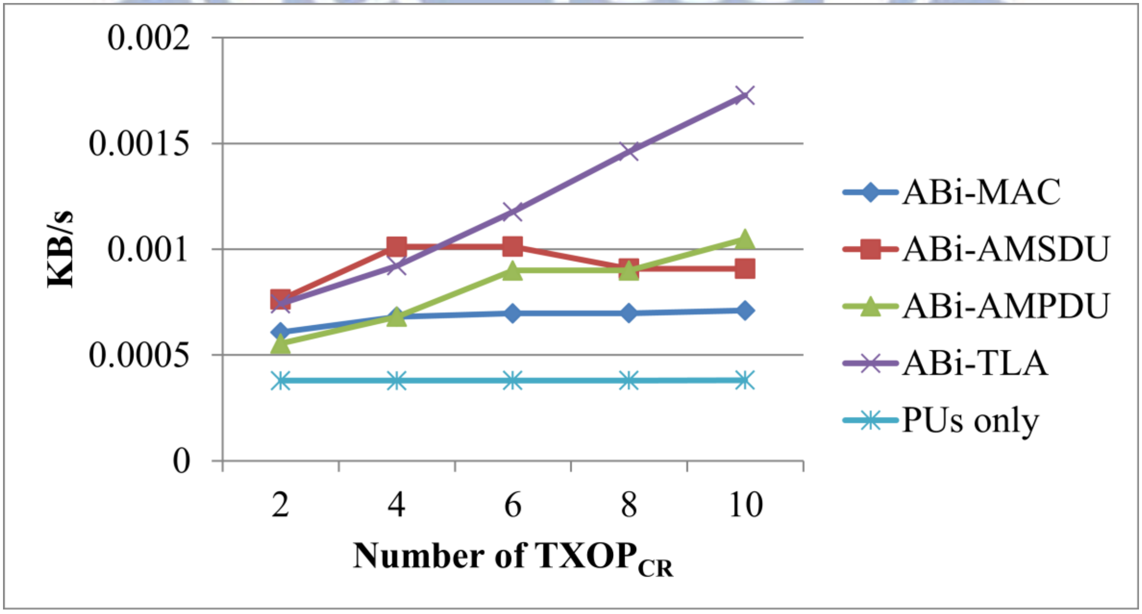


Figure 6.10 PU's Average Packet Delay Time (30% PU Load)

Table 6.11 The PU's Throughput of 30% Load with the effect of CRUs (KB/s)

	$TXOP_{CR}$ = 2	$TXOP_{CR}$ = 4	$TXOP_{CR}$ = 6	$TXOP_{CR}$ = 8	$TXOP_{CR}$ = 10
ABi-MAC	1050.49	1050.80	1050.61	1050.64	1050.64
ABi-AMSDU	1050.64	1050.46	1050.77	1050.64	1050.58
ABi-AMPDU	1050.58	1050.49	1050.43	1050.49	1050.43
ABi-TLA	1050.46	1050.46	1050.40	1050.37	1050.34

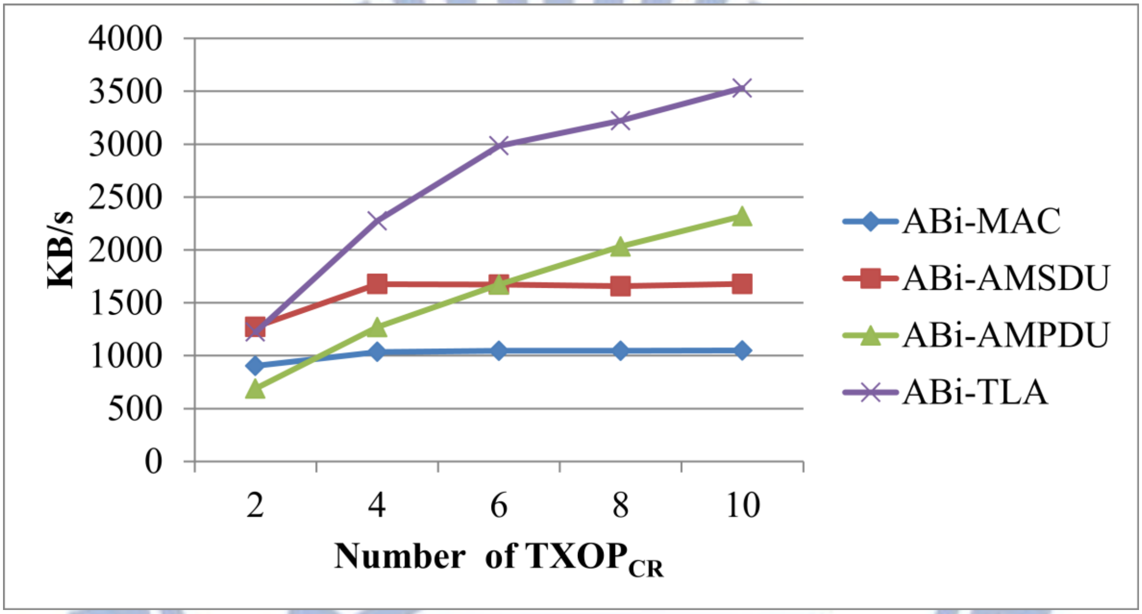


Figure 6.11 TCP Throughput of CRU with 30% PU Load

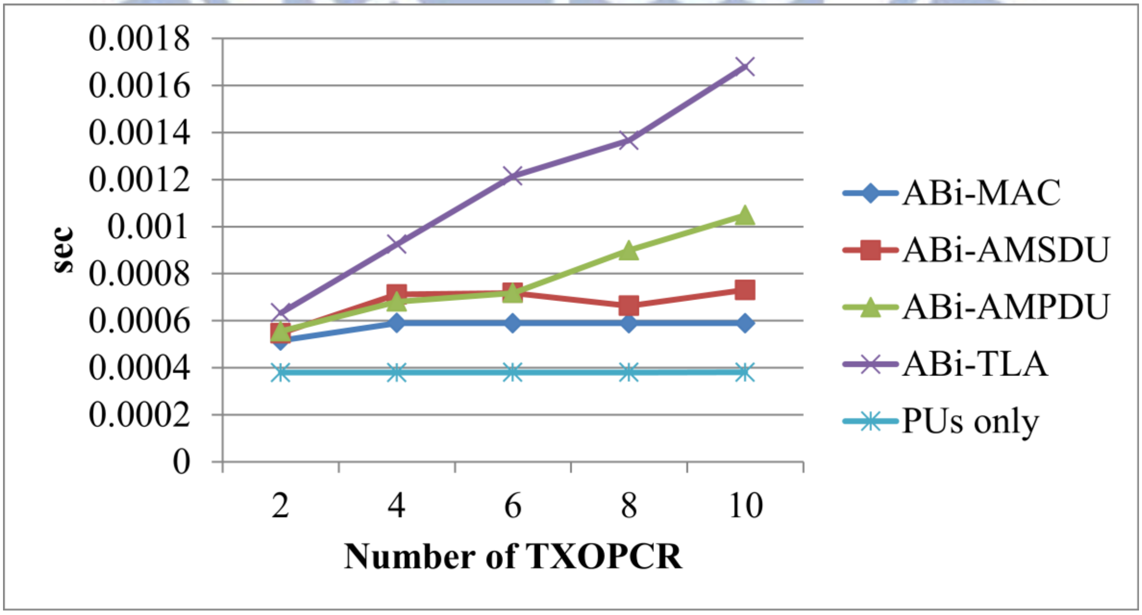


Figure 6.12 PU's Average Packet Delay Time (30% PU Load)

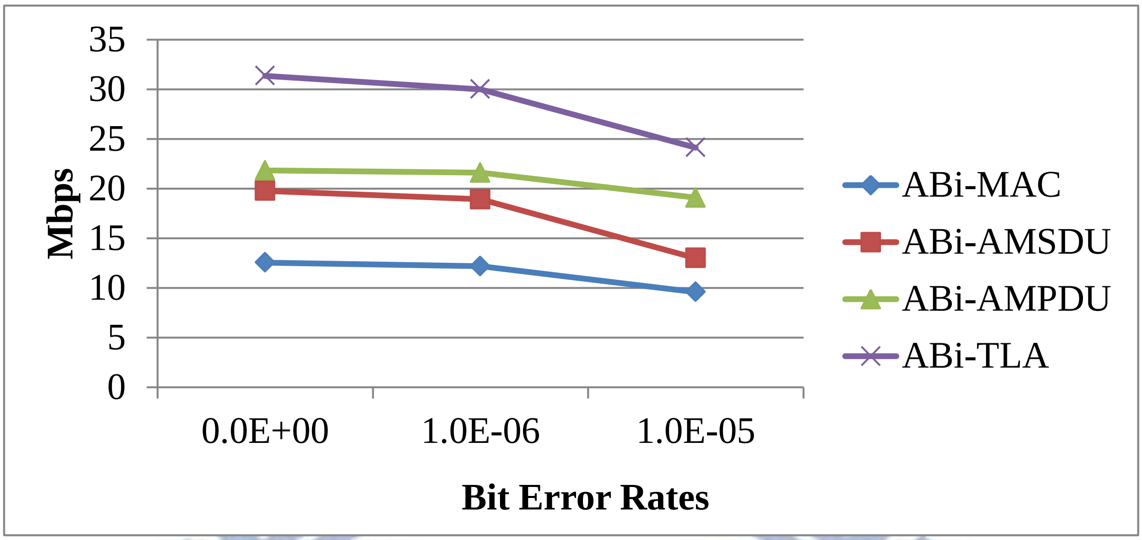


Figure 6.13 Application Layer UDP Throughput of ABi-MAC with Different Bit Error Rates

(With UDP payload 1450 bytes, $TXOP_{CR}$ = 10)

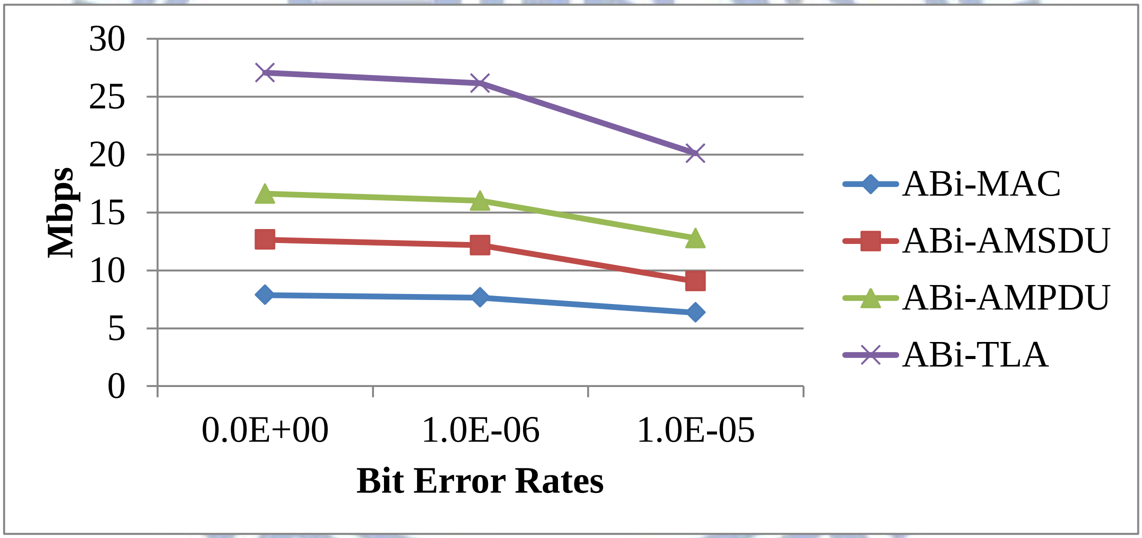


Figure 6.14 Application Layer TCP Throughput of ABi-MAC with different Bit Error Rates

($TXOP_{CR}$ = 10)

Figure 6.13 and Figure 6.14 shows the UDP/TCP throughput of different FA-AMCs under different wireless spectrum condition. We only show the result when $TXOP_{CR}$ is 10 since each FA-MAC can get the best performance with this setting. On can see the influence to the throughput from different BERs is same as the result in case one. The FA-MACs applies with A-MSDU aggregation suffers more throughput degradation.

- **ABi-MAC with 50% PU load**

When PUs are creating 50% load, the effect from CRUs is almost the same as PU creating 30% load.

In Table 6.12, we can see that the throughput of PU still remain the same at about 1,841 KB/s with the influence of CRUs. In Figure 6.15, the throughput of CRU pair still benefit from frame aggregation mechanism like the way we mentioned before at section 6.4. First, the improvement of A-MPDU is obvious when comparing the curves of ABi-MAC and ABi-AMSDU. However, the throughput difference between $TXOP_{CR}$ value 2 and 4 is become less than it does in Figure 6.9. This is because the average packet generation interval of 50% load is about only 0.9 ms (see the setting in Table 6.4). In average, the change that CRU overhears the channel claiming request from PU at the second data transmission round is bigger. This is the reason why the throughput difference between $TXOP_{CR}$ value 2 and 4 is less than 30% PU load.

Second, the improvement of ABi -AMPDU aggregation grows bigger while $TXOP_{CR}$ is bigger and causing bigger PU’s packet delay time. However, maximum throughput is less than the 30% PU load case due to more channel access time is occupied by PUs.

The situation is the same when CRU is applying TCP flow. First, the throughput of PU still remains the same at about 1,841 KB/s with the influence of CRUs in Table 6.13. Second, the throughput difference between $TXOP_{CR}$ values equal to 2 and 4 of ABi-MAC and ABi-AMSDU is less. Last, ABi-AMPDU and ABi-TLA cannot achieve the high throughput as in 30% PU load case.

Table 6.12 The PU's Throughput of 50% Load with the effect of CRUs (KB/s)

	$TXOP_{CR}$ = 2	$TXOP_{CR}$ = 4	$TXOP_{CR}$ = 6	$TXOP_{CR}$ = 8	$TXOP_{CR}$ = 10
ABi-MAC	1842.053	1841.81	1841.81	1841.901	1841.566
ABi-AMSDU	1841.901	1841.81	1841.81	1841.201	1841.323
ABi-AMPDU	1841.901	1841.901	1841.688	1841.688	1840.197
ABi-TLA	1841.688	1841.688	1840.41	1840.866	1837.731

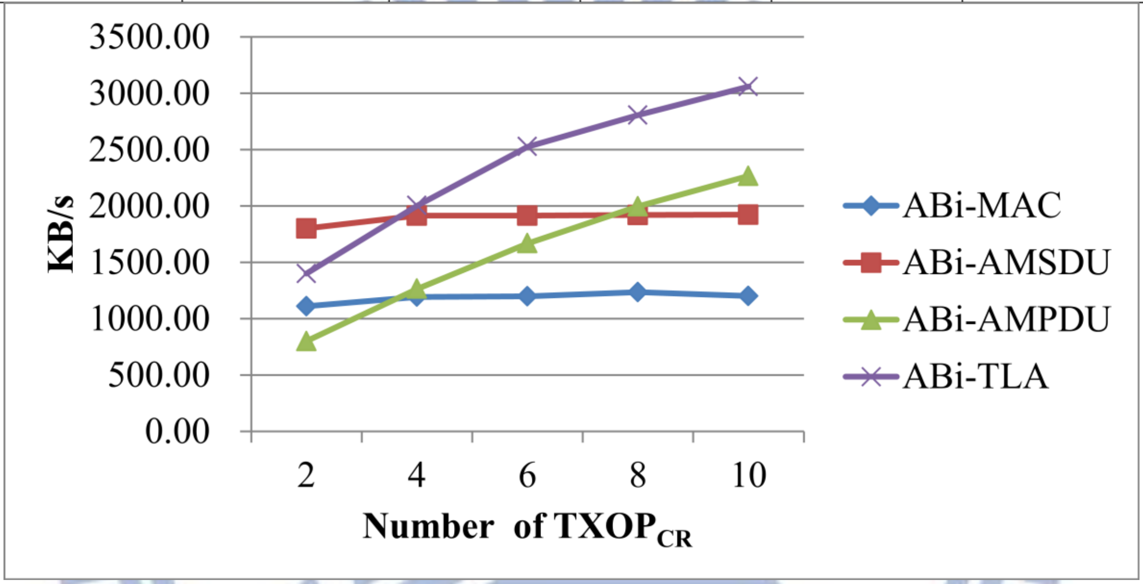


Figure 6.15 UDP Throughput of CRU with 50% PU Load

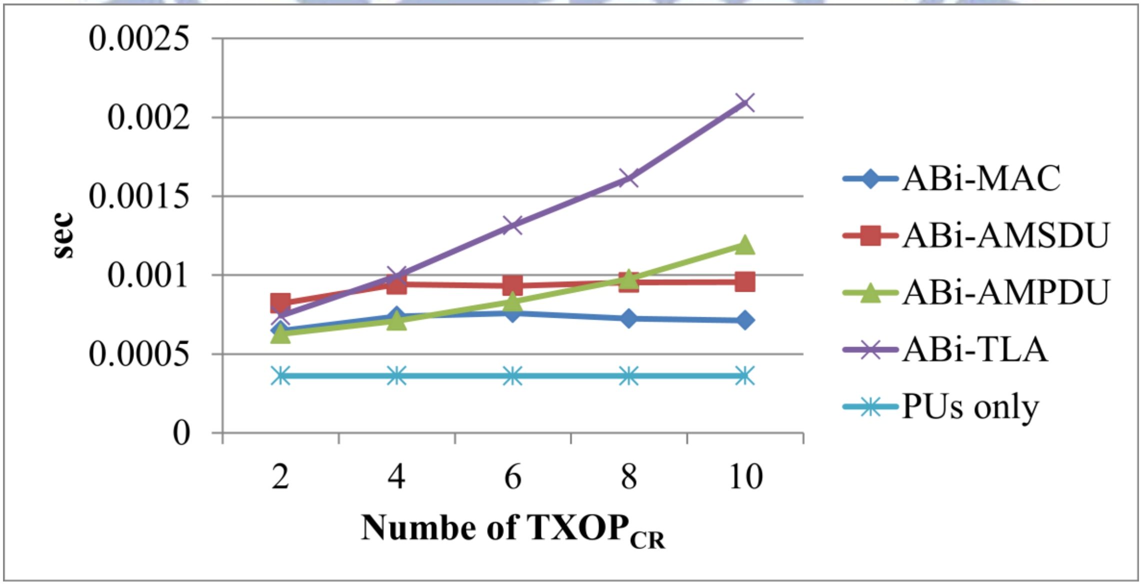


Figure 6.16 PU's Average Packet Delay Time (50% Load)

Table 6.13 The PU's Throughput of 50% Load with the effect of CRUs (KB/s)

	$TXOP_{CR}$ = 2	$TXOP_{CR}$ = 4	$TXOP_{CR}$ = 6	$TXOP_{CR}$ = 8	$TXOP_{CR}$ = 10
ABi-MAC	1841.749	1841.749	1841.688	1841.536	1841.901
ABi-AMSDU	1841.749	1841.719	1841.262	1841.81	1841.445
ABi-AMPDU	1841.749	1841.81	1841.688	1841.505	1841.171
ABi-TLA	1841.871	1840.653	1841.14	1841.566	1838.705

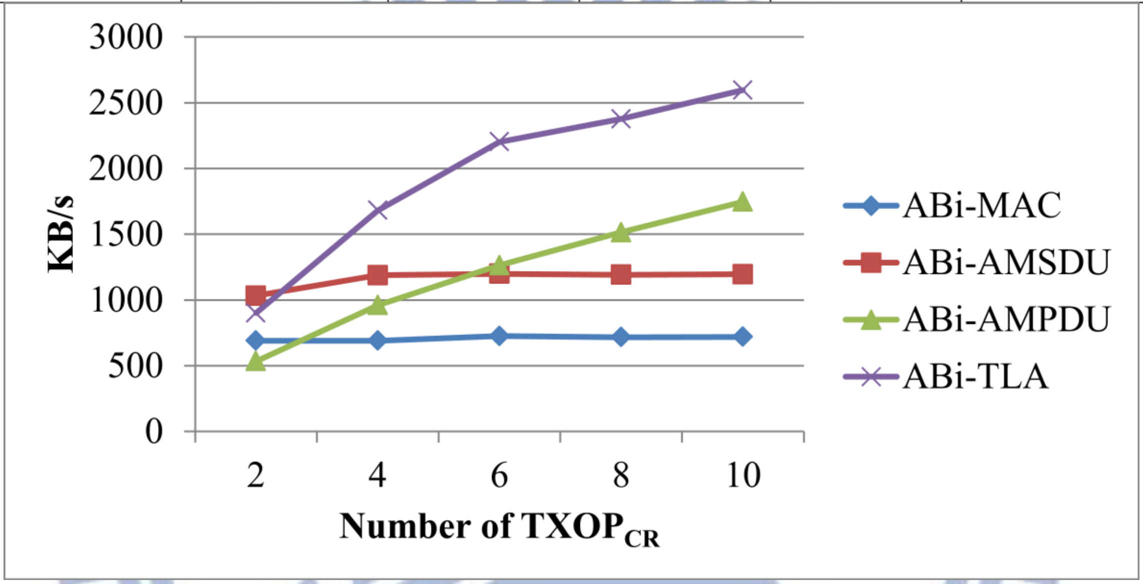


Figure 6.17 TCP Throughput of CRU with 50% PU Load

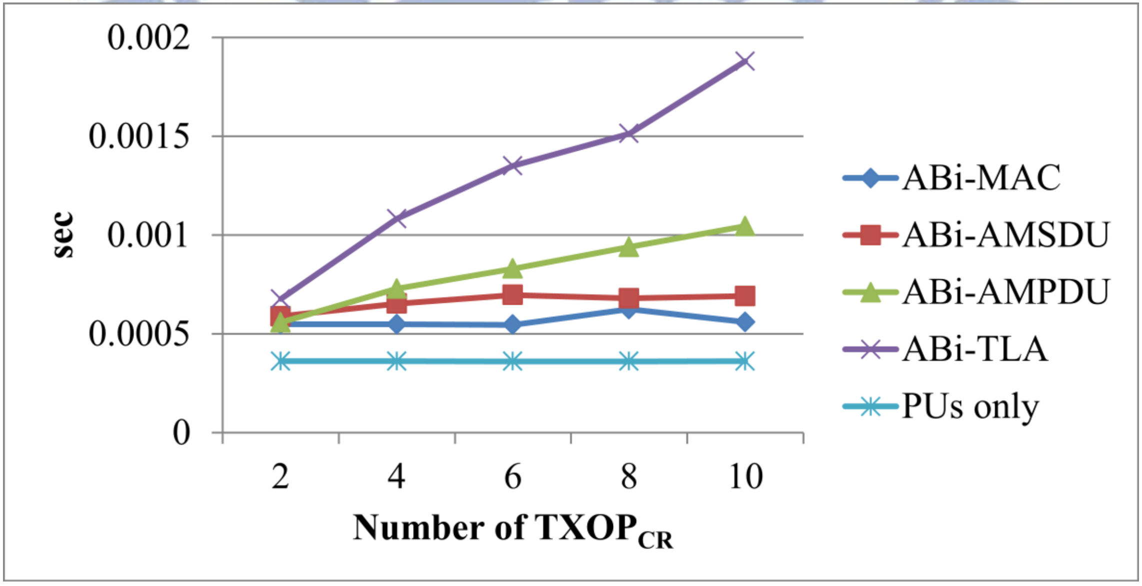


Figure 6.18 PU's Average Packet Delay Time (50% Load)

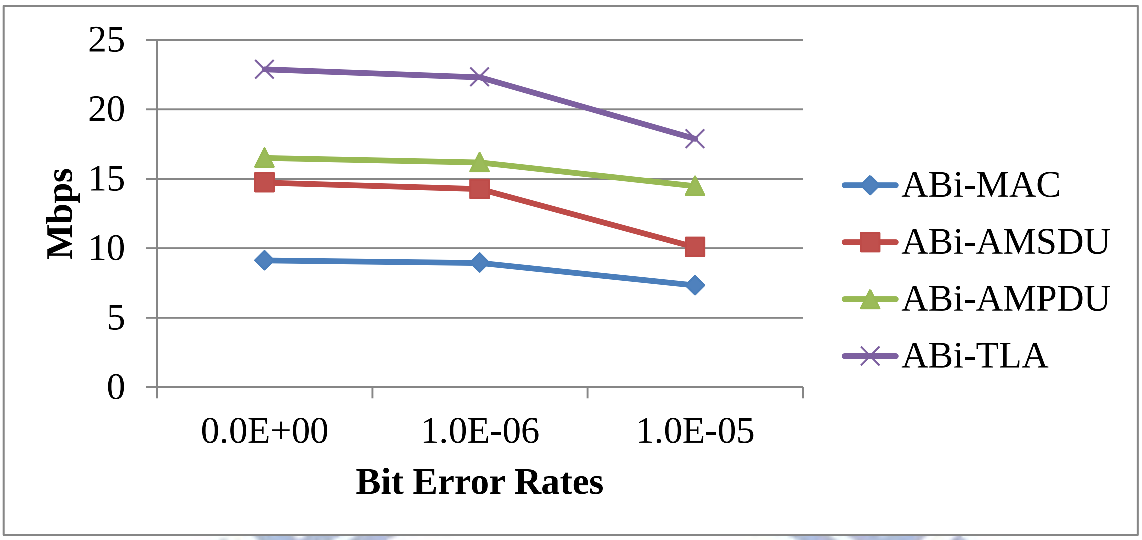


Figure 6.19 Application Layer UDP Throughput of ABi-MAC with different Bit Error Rates

(UDP payload 1450 bytes, $TXOP_{CR}$ = 10, 50% PU Load)

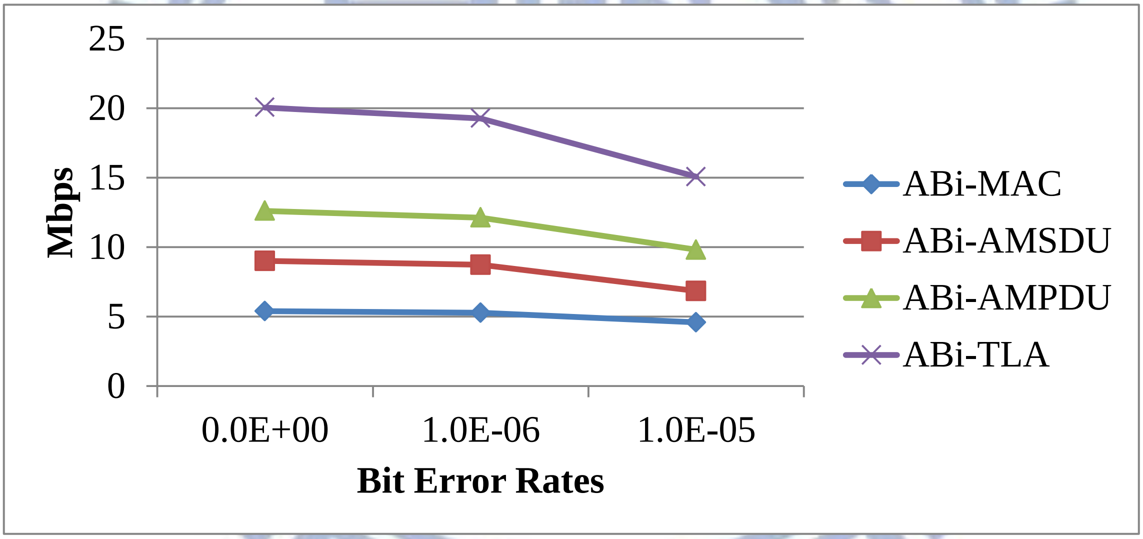


Figure 6.20 Application Layer TCP Throughput of ABi-MAC with different Bit Error Rates

($TXOP_{CR}$ = 10, 50% PU Load)

Figure 6.19 and Figure 6.20 show UDP/TCP throughput of different FA-AMCs under different wireless spectrum condition. On can see the influence to the throughput from different BERs is same as the result in case one. The FA-MACs applies with A-MSDU aggregation suffer more throughput degradation.

- **ABi-MAC with 80% PU load**

In this case, the throughput of PUs is influenced by CRU when any frame aggregation mechanism is applied as shown in Figure 6.21. The throughput is even influenced by ABi-MAC with no frame aggregation mechanism is applied. This is because the average packet interval of 80% PU load is about 500 us. Since we change the sensing time of CRU to 200 us only, it is still possible for CRUs to access the channel. On can see the PU's packet delay increase about 2 ms. However, this will decrease the throughput of PUs. In both TCP and UDP traffic flow. For the worst case: ABi-TLA causes the biggest influence after $\mathrm{TXOP}_{\mathrm{CR}}$ is bigger than 4 due to the Tow Level Aggregation.

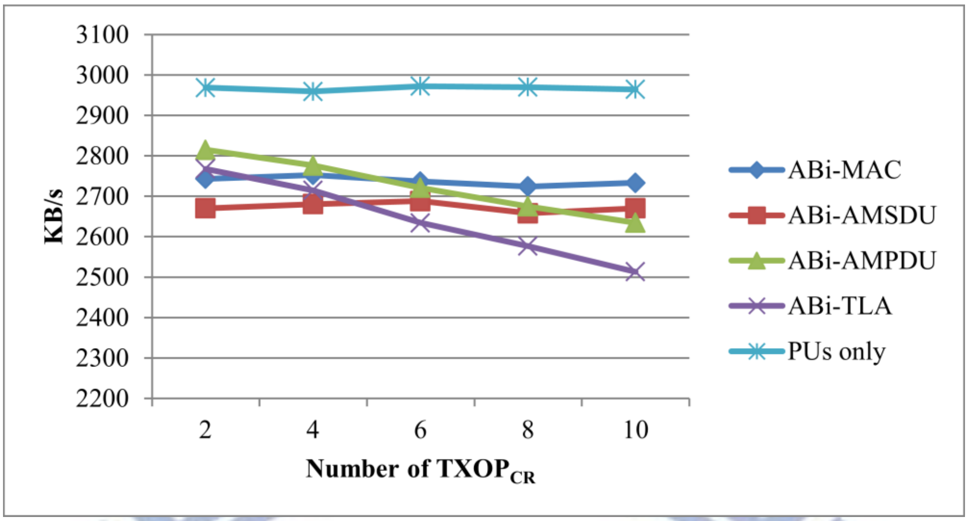


Figure 6.21 The Throughput of PUs with the effect of CRUs (80% PU Load)

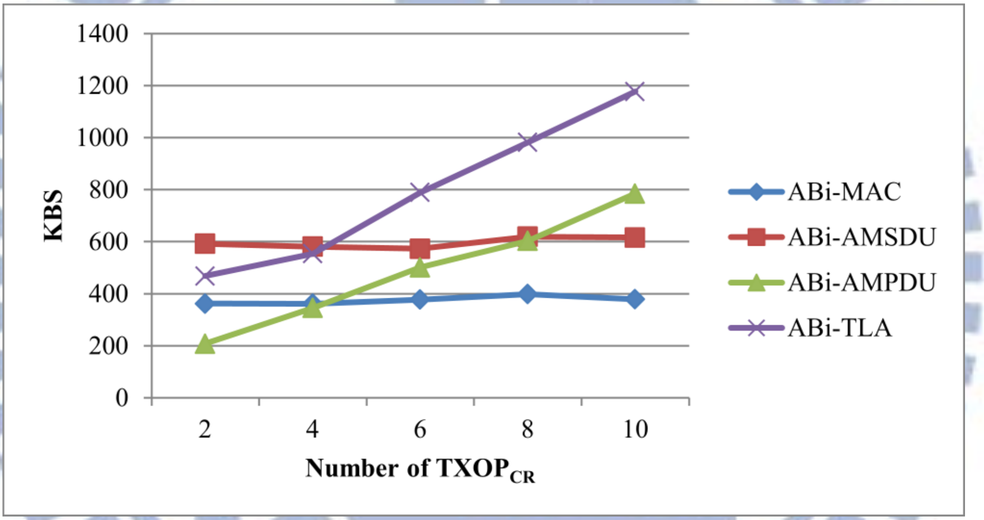


Figure 6.22 UDP Throughput of CRU with 80% PU Load

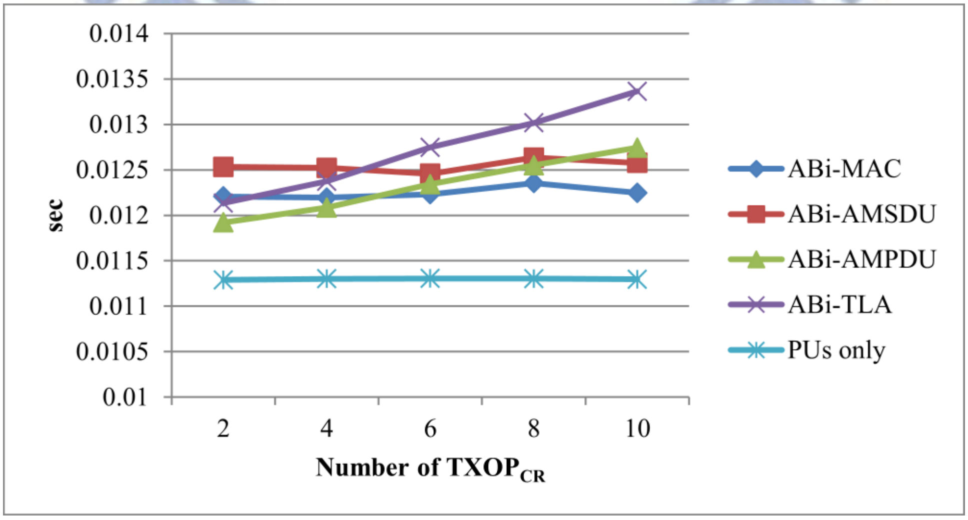


Figure 6.23 PU’s Average Packet Delay Time (80% PU Load)

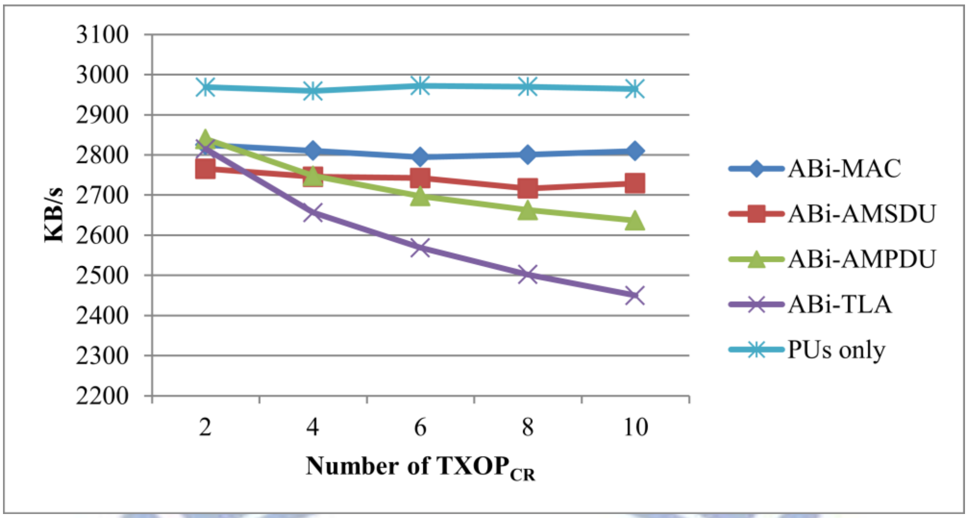


Figure 6.24 The Throughput of PUs with the effect of CRUs (80% PU Load)

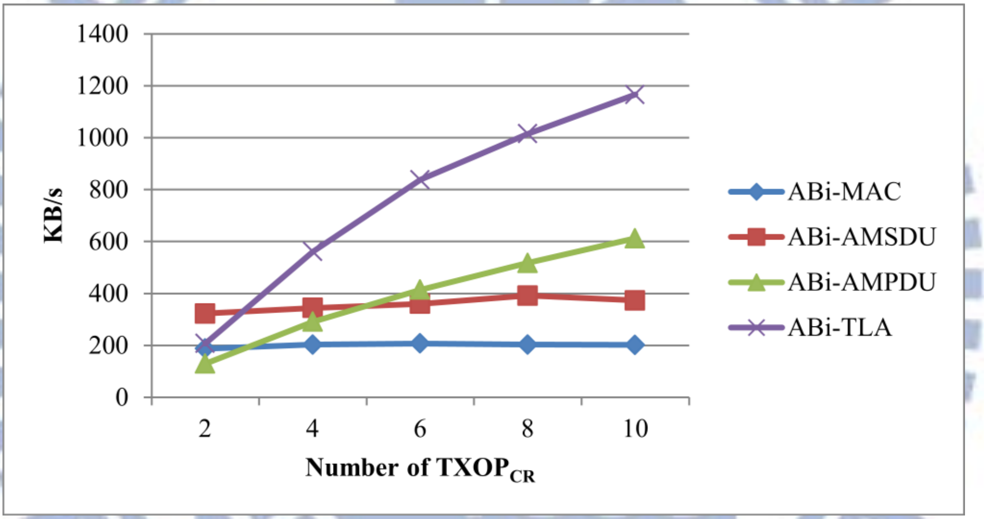


Figure 6.25 TCP Throughput of CRU with 80% PU Load

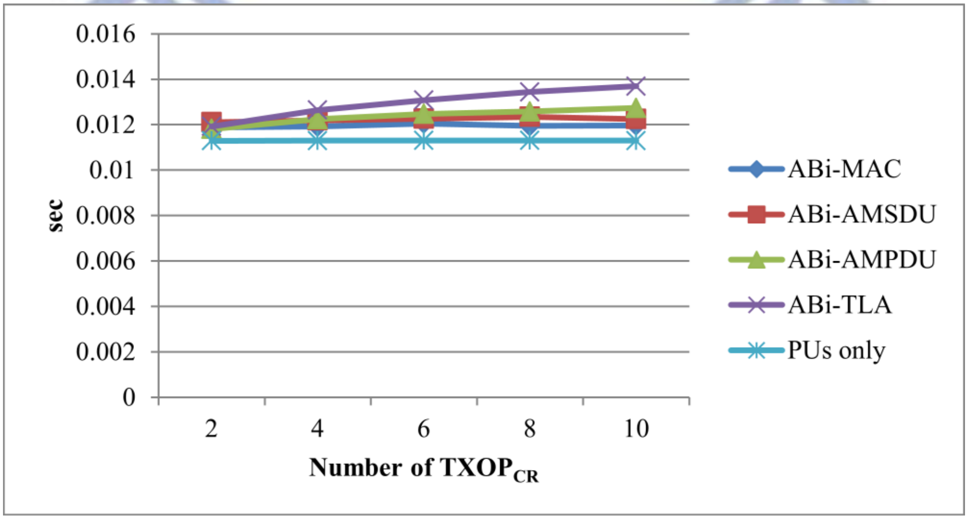


Figure 6.26 PU's Average Packet Delay Time (80% PU Load)

6.4. Summary

In this chapter, we evaluate the influence of CRU to PU under different channel busy degree. One can see that in the cases of 30% PU load and 50% PU load, the throughput of PUs do not influence by CRUs when different frame aggregation mechanism is applied. However, the CRUs do increase the packet delay time of PUs. The worst case is in Figure 6.16 caused by ABi-TLA with two-way UDP flow. Even though the packet delay time is increased, the range of increment is less than 2 milliseconds that is acceptable for the real time application such as video streaming and voice call over IP. We think the influence of packet delay time is tolerable for PUs. However, CRUs do make a huge influence to PUs in the case of 80% PU load. The reason is the channel sensing time that we modify it to only 200 us. The reduced channel sensing time can increase the efficiency of CR-MAC protocol but break the protection to PUs under busy channel status. Fortunately, this situation happens only when every data channel is all busy. Otherwise, the CRUs will avoid accessing the busiest channel due to SCSS.

Under the consideration of BER, the simulation result shows that using A-MSDU aggregation mechanism may cause bigger performance degradation. However, the benefit is still greater than the drawback while using it.

Chapter 7 Conclusion

In this dissertation, we first migrate the former 802.11 Cognitive Radio Network protocol from old NCTUns network simulator to new Estinet network simulator and evaluate them with a higher data rate and new version of IEEE 802.11a PUs. We also propose New Dynamic Bandwidth Allocation and New Bandwidth Negotiation Process to enhance the former CR MAC protocol. Then we implement different frame aggregation mechanisms on CR MAC protocol in order to improve the performance. In chapter 6, we design the simulation cases to evaluate the performance of CR MAC protocol applied with different frame aggregation mechanism. The result shows that frame aggregation mechanism do enhance the performance of CRUs. The effect of Two Level Aggregation is the best if $TXOP_{CR}$ setting is big enough. The A-MSDU aggregation has an instault improvement with small $TXOP_{CR}$ values, i.e., 1 or 2. Through three different PU load cases, we realized that proposed FA-MAC can operate correctly and more efficiently with the existence of PUs. However, the FA-MAC shows its drawback in the situation that every data channel is very busy.

To solve the current drawback above, a more dynamic channel sensing time and $TXOP_{CR}$ settings can be applied to make FA-MAC more intelligent so that it can fit into different network situations. Through controlling the applied frame aggregation mechanism and $TXOP_{CR}$ setting, CR MAC protocol can perform in a more efficient and less PU-effected way. However, the current channel sensing mechanism with only antennas cannot provide enough information to decide the most suitable frame aggregation mechanism and $TXOP_{CR}$ setting. Since many wireless devices are using multiple antennas and the technique of multi-antenna has become cheaper and more mature. In the future, more advanced channel sensing mechanism using multiple antennas is a good way to improve the CR MAC protocol.

Chapter 8 REFERENCES

[1] H. TakChou Lou and Tzu-Jane Tsai, “On Synchronized Channel Sensing and Accessing for Cognitive Radio Users in IEEE 802.11 Wireless Networks,” in PIMRC 2009, Tokyo, Japan, 13-16 September 2009.

[2] Shie-Yuan Wang, Yu-Ming Huang, Lee-Chin Lau and Chih-Che Lin, "Enhanced MAC Protocol for Cognitive Radios over IEEE 802.11 Networks", IEEE Wireless Communication and Networking Conference 2011 (WCNC 2011), 28-30 March 2011, Cancun, Mexico.

[3] Lee-Chin Lau, Chih-Che Lin and Shie-Yuan Wang, “Bi-directional Cognitive Radio MAC Protocol For Supporting TCP Flows,” accepted by IEEE 74th Vehicular Technology Conference (VTC 2011), September 5-8 2011, San Francisco, United States.

[4] L.C. Lau, C.C. Lin, and S.Y. Wang, “An IEEE 802.11 Cognitive Radio MAC Protocol with Dynamic Bandwidth Allocation Capabilities,” IEEE WCNC 2012 (Wireless Communications and Networking Conference 2012), April 1 - 4, 2012, Paris, France.

[5] Y. Xiao, “IEEE 802.11n: Enhancements for Higher Throughput in Wireless LANs,” IEEE Wireless Communications, vol. 12, no. 6, pp. 82–91, Dec. 2005.

[6] D. Skordoulis, Q. Ni, H.-H. Chen, A. Stephens, C. Liu, and A. Jamalipour, “IEEE 802.11n MAC frame aggregation mechanisms for next generation high-throughput WLANs,” Wireless Communications, IEEE, vol. 15, no. 1, pp. 40–47, February 2008.

[7] IEEE 802.11 Working Group, “IEEE 802.11-2007: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”, June 2007.

[8] IEEE 802.11n-2009: Standard for Information Technology - Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks - Specific Requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Amendment 4: Enhancements for Higher Throughput. IEEE Std P802.11n, October 2009.

[9] J. Mitola and G. Q. Maguire, Cognitive radio: Making software radios more personal, IEEE Personal Communications, vol. 6, no. 4, pp. 1318,Aug. 1999.

[10] IEEE 802.22 Working Group on Wireless Regional Area Networks Enabling Rural Broadband Wireless Access, Available: http://ieee802.org/22.

[11] Standard ECMA-392 MAC and PHY for Operation in TV White Space, Available: http://www.ecmainternational.org/publications/standards/Ecma-392.org/11.

[12] "IEEE 802.15 wpan task group 2 (tg2)," http://www.ieee802.org/15/pub/TG2.html

[13] M. A. McHenry, NSF spectrum occupancy measurements project summary, shared spectrum co. report, Aug. 2005.

[14] Estinet network simulator is available at http://www.estinet.com/.

[15] NCTU network simulator is available at http://nsl.csie.nctu.edu.tw/nctuns.html