國立交通大學

資訊科學與工程研究所

碩士論文

在 IEEE 802.11 網路上
具有動態頻寬配置功能的
雙向感知無線電媒介層多重存取協定

Bi-directional Cognitive Radio MAC Protocol with Dynamic Bandwidth Allocation over IEEE 802.11 Networks

研 究 生：劉莉晶
指導教授：王協源　教授

中華民國一百年六月

在 IEEE 802.11 網路上具有動態頻寬配置功能的 雙向感知無線電媒介層多重存取協定

Bi-directional Cognitive Radio MAC Protocol with Dynamic Bandwidth Allocation over IEEE 802.11 Networks

研究生：劉莉晶　　Student： Lee-Chin Lau

指導教授：王協源　　Advisor：Shie-Yuan Wnag

國立交通大學
資訊科學與工程研究所
碩士論文

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

Computer Science

June 2011

Hsinchu, Taiwan, Republic of China

中華民國一百年六月

Bi-directional CR MAC Protocol with Dynamic Bandwidth Allocation over IEEE 802.11 Networks

Student: Lee-Chin Lau Advisor: Dr. Shie-Yuan Wang

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

ABSTRACT

Recently, the CR technology has emerged as a recent breakthrough radio technology to address the spectrum underutilization and spectrum scarcity problem by providing means to utilize the spectrum holes (i.e., unused spectrum) in an intelligent way. Therefore, this technology could facilitate efficient utilization of the radio spectrum while provide highly reliable communication for all users of the targeted wireless networks. In this research, we propose a bi-directional CR MAC protocol for the CR users (CRUs) such that the CRUs can coexist with the primary users (PUs, i.e., IEEE 802.11 users) and flexibly use the spectrum holes without affecting the performances of PUs.

The proposed bi-directional CR MAC protocol is an extension of the CR MAC protocol proposed by TakChon Lou et al. [9] and Shie-Yuan Wang et al. [10]. Although both of these previous works have dramatically improved the spectrum efficiency, their studies fail to address the protocol degradation performance in delivering TCP flows. In addition, the main limitation of their protocols is that the bandwidth allocation for the CRUs has to be done statically. Therefore, in this research, we first address the problem of the TCP performance degradation over [10] and propose a basic scheme CR MAC protocol (BBi-MAC) as a remedy. Follow on, we work on an advanced scheme of the CR MAC protocol (ABi-MAC) which possesses the ability of reserving the bandwidth dynamically according to the CRUs' bandwidth demand. Therefore, the ABi-MAC can further well utilize the spectrum holes of an IEEE 802.11 network and support synchronous two-way traffic flows with variable packet sizes. The performances of our proposed schemes are evaluated using both mathematical analysis and simulation. Both of the analytical and numerical results show that our proposed schemes can significantly enhance the aggregate throughputs of CRUs in an 802.11 network and dramatically improve the spectrum efficiency without degrading the performances of PUs.

Keywords: Cognitive Radio, 802.11, TCP, dynamic bandwidth allocation

致謝

過去從沒想過自己會來到臺灣讀書。回顧兩年前，毅然放棄澳州大學博士候選人的身份，決定來臺灣交通大學重新從碩士班開始裝備自己。我非常慶幸自己當初作了這個決定。兩年的交大研究之旅雖然短暫，卻已成為我人生最重要的轉捩點。感謝指導教授王協源老師一路來悉心的栽培及教導，讓我的研究、思考，以及解決問題的能力都逐漸成熟。我也要感謝我的校內外論文指導委員（趙禧祿老師、伍紹勳老師以及清華大學林華君老師）給予我的論文具有建設性的建議，使我能在畢業之際發掘自己的缺點，加以改進，成為一個更成熟的研究者。我亦要感謝所有實驗室裡的學長學弟妹，因為你們的陪伴，讓我在臺灣的生活增添了不少的色彩和歡笑。

另外，我特別要感謝我媽咪無私的愛、支持與體諒。感謝媽咪在爸爸去世後，仍舊堅強的把我們四個小孩撫養成人，並悉心的栽培我們。謝謝媽咪在我升大學至現在碩士畢業的這段漫長就學期間，成為我生活上隨時的後盾。謝謝媽咪在我做每一個大小決定的時候，給予我許多有智慧的建議並支持。沒有您，就沒有今天的我。

我亦要感謝我的男友母親，在異地求學時非常地照顧我，給與我很多關心與生活上的照顧，讓我在台灣也能有家的感覺。最後，我要感謝我的學長兼男友志哲。謝謝你一路來的陪伴和指導，讓我在研究的路上，從一開始的摸索、彷徨，到如今的享受研究。謝謝你讓我有機會來到臺灣、認識臺灣。以後將一起在這片土地上，攜手共度此生。

劉莉晶　撰於　民國一百年六月二十九日

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The success of wireless technology today has motivated the explosion of new wireless applications which creates an ever-increasing demand for more radio spectrum. However, most of those easily accessed spectrum band have been statically allocated. This is because with the fixed spectrum allocation policies, the bandwidth spectrum used by different radio technologies are fixed and regulated by the authorities of each country (e.g., Federal Communications Commission of the United States and National Communications Commission of Taiwan). Therefore, each statically allocated frequency band shall be accessed only by specific authorized users as shown in Fig. 1.1 and Fig. 1.2.

However, recent studies [1][2] have shown that under fixed spectrum allocation policies, most of the radio frequencies were inefficiently utilized. For example, from a measurement study conducted by FCC [1] in year 2002, it is found that at any given time and in any geographic locality, less than 10% of the available spectrum is being utilized. The FCC also highlighted that a large portion of the licensed spectrum is used sporadically and that geographical variations in the utilization of licensed spectrum portions oscillates from 15% to 85% with high variance in time. This report challenges for the first time the

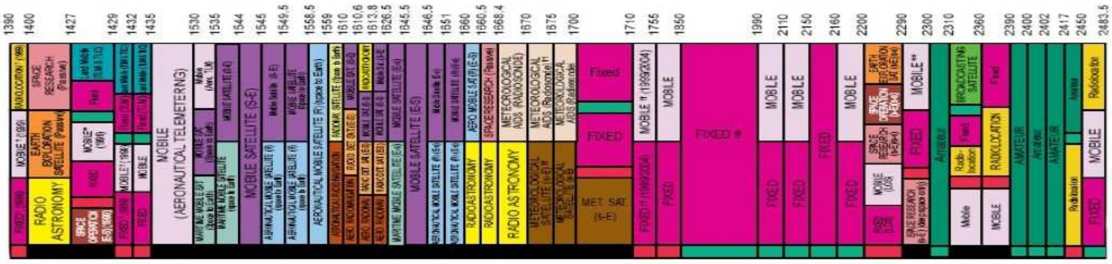


Figure 1.1: Illustration of the Midband Spectrum Occupation

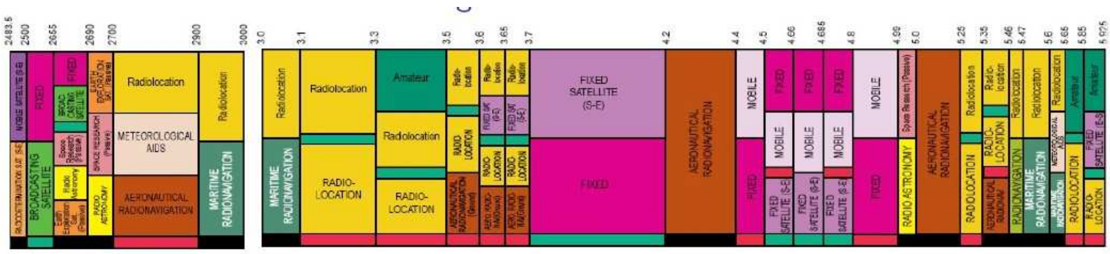


Figure 1.2: Illustration of the Highband Spectrum Occupation

common belief of spectrum scarcity. To exploit underutilized portions of the spectrum (a.k.a., white spaces, spectrum holes and etc.), FCC recommended the deployment of wireless devices that can coexist with the licensed users (also called authorized users) to achieve a significantly greater spectral efficiency. Such findings and recommendation have motivated the search for breakthrough radio technologies that can scale to meet future demands both in terms of spectrum efficiency and application performance.

Cognitive Radios (CRs) thus emerged as a new generation of radio technology that will enable the future wireless world. CRs are smart, fully programmable radios that are capable of interference sensing, environment learning, and dynamic spectrum access. With these capabilities, a CR can adapt its transmission or reception parameters to communicate efficiently avoiding interference with licensed or unlicensed users. Therefore, it is anticipated that the CR technology can increase the bandwidth utilization of the radio spectrum in a global perspective while stimulate the deployment of new wireless applications.

The idea of CR was first presented by Dr. Joseph Mitola III [3][4] which is generally an evolved software-defined radio [5]. As compared to the software-defined radios, CR is smarter and possesses more intelligence through machine learning. The importance of this technology has been affirmed through the emergence of CR-related standards, namely the IEEE 802.22 [6] and ECMA-392 [7]. Both of these standards define the MAC and PHY operations in TV white space so that the unused TV channels can be utilized for data communications, provided that the licensed users' operations can be protected. This type of CR is regarded as licensed band CR such that the CRs can use those spectrum bands that have been assigned to licensed users. This kind of spectrum sharing is also referred to as vertical sharing, as indicated in Fig. 1.3. CR operating in frequency bands of TV and radio broadcasts is a typical example of vertical spectrum sharing. Fig. 1.4 further

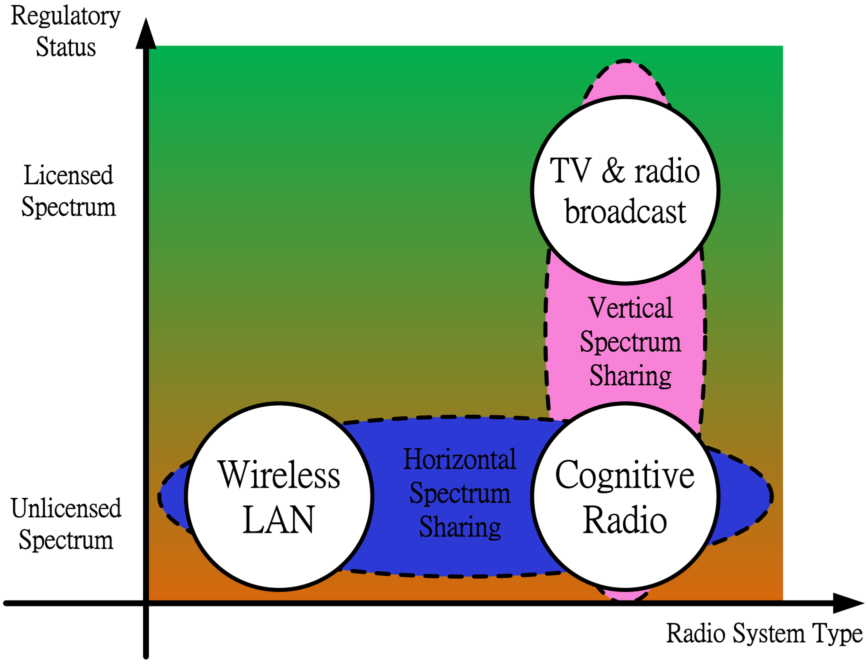


Figure 1.3: Illustration of the Vertical and Horizontal Spectrum Sharing

illustrates an example of vertical spectrum sharing such that CRs at different locations can identify different frequencies (i.e., TV frequency band I and TV frequency band II) as unused and regard them as spectrum opportunities.

Contrarily, unlicensed band CR is another type of CR which the CRs can only utilize unlicensed parts of radio spectrum, for example, the spectrum of the IEEE 802.11 networks. This kind of spectrum sharing is referred to as horizontal sharing as shown in Fig. 1.3. For horizontal spectrum sharing, the CRs are with limited coexistence capabilities, enabling them to operate in spite of some interference from dissimilar radio

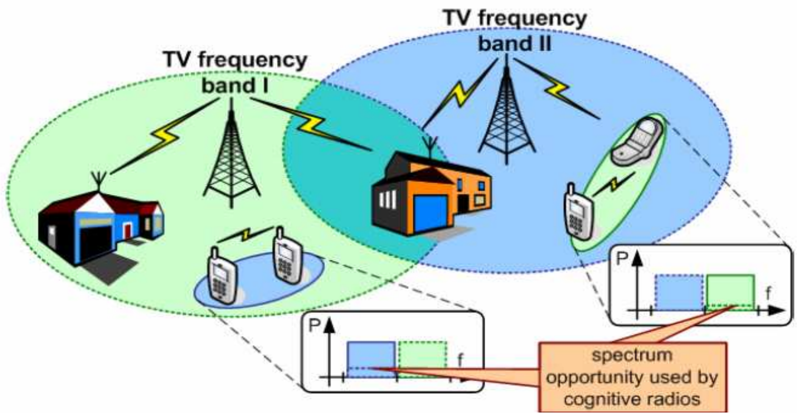


Figure 1.4: Example of the Vertical Spectrum Sharing

systems.

The analysis of the network traffic done by Yingxi Liu et al. [8] showed that unsaturated 802.11 networks stay idle as much as 50% of the time. This fact has hugely motivated proposals of CR MAC protocol over the 802.11 networks to form unlicensed band CR networks (CRNs). In such an 802.11 CRN, primary users (PUs) refer to 802.11 nodes authorized to use this frequency band. Contrarily, CR users (CRUs) are 802.11 nodes equipped with CRs who are unauthorized to use this frequency, hence could only access the 802.11 spectrum in an opportunistic manner without interfering with the PUs.

1.1 Problem Description

This thesis is an extension of the work done by [9] and [10]. TakChon Lou et al. [9] proposed a MAC protocol that allows CRUs to access the idle period of an IEEE 802.11 network opportunistically. Furthermore, these CRUs can flexibly abort their data transmissions once they detect that PUs intend to access the authorized frequencies. Shie-Yuan Wang et al. [10] further expanded the work done by [9] through proposing a Smart Channel Selection Scheme (SCSS) and addressing the problem of Hidden Terminals (HTs). Through extensive simulations, Shie-Yuan Wang et al. found that the Transmission Control Protocol (TCP) did not perform well under their extended work.

Since TCP has become that most widely used transport-layer protocol today, it is imperative that a proposed CR MAC protocol can well support TCP flows. Therefore, in this work, we first address the problem of the TCP performance degradation over [10] and propose a basic scheme of bi-directional CR MAC protocol (BBi-MAC) as a remedy. Follow on, we work on an advanced scheme of the bi-directional CR MAC protocol (ABi-MAC) with dynamic bandwidth allocation to support asymmetry two-way traffic flows.

1.2 Dissertation Organization

The remainder of this dissertation is organized as follows. In Chapter 2, we first present the related works. Specifically, we introduce the CR MAC protocol which was originally proposed by [9], then modified for better performance by [10]. Then, we address the problem of these related works in supporting TCP and bi-directional traffic flows. In Chapter 3, we explain both the proposed basic scheme and advanced scheme of bi-directional CR MAC protocol for supporting bi-directional and asymmetry traffic flows, respectively. In Chapter 4, the performances of the proposed bi-directional CR MAC protocols are evaluated using mathematical analysis and simulations and we discuss the achieved enhancements by comparing them to [9]. We finally conclude this dissertation and present our future work in Chapter 5.

Chapter 2

Background and Related Work

2.1 Introduction to Cognitive Radio Network

CR has emerged as a new design paradigm for next-generation wireless networks. It is also regarded as a recent breakthrough radio technology which can greatly increase utilization of the scarce radio spectrum (both licensed and unlicensed frequency) and to enable new wireless applications. It was observed that some frequency bands in the radio spectrum are largely unused, while some are heavily used. In particular, while a frequency band is assigned to a primary wireless system at a particular time and location, the same frequency band is unused by this wireless system in other times and locations. This results in spectrum holes (also called spectrum opportunities). Therefore, the spectrum utilization can be improved substantially by allowing secondary users, i.e., CR users, to utilize these spectrum holes.

The CR technology is evolved from the concept of Software Defined Radio (SDR) which was introduced to enhance the efficiency of frequency spectrum usage [5]. SDR improves the capability of a wireless transceiver by using embedded software to enable the radio transceiver to operate in multiple frequency bands. The CR, however, is a special type of SDR which is able to intelligently adapt itself to the changing environment. Therefore, learning and adaptation are two significant features of a CR transceiver.

In general, the major four components of a CR system includes: observation, learning, decision making and planning, and acting processes. The details of these components are elaborated as below:

- Observation Process – The observation process typically consists of measurements and noise reduction mechanisms. It can be either passive or active. In a passive observation approach, a CRU silently listens to the environment. In contrast, in an active observation approach, a CRU measures the nearby signal level and transmits a special message. Through exchanging these special messages, CRUs can be updated on the information about the surrounding environment.

- Learning Process – This refers to the process of extracting useful information from collected data or messages. A learning process utilizes data from the observation process, and previous decisions and actions. For example, a CR sender can learn the network behavior (e.g., through interference or collision in PHY and MAC layers, respectively) to gain knowledge of the operating environment. There are two major types of learning algorithms, supervised learning and reinforcement learning. In a supervised leaning algorithm, a large set of training examples with known solutions are used to train the algorithm. This type of algorithm is more suitable for offline operation where sample data is available. On the other hand, the reinforcement learning algorithm learns by interacting with th environment. It is used when the correct solution is unknown. A reinforcement learning algorithm tries different actions and observes the outcomes. The outcomes information is then used to optimize the action in the future. Because the actions through this learning algorithm can be adapted dynamically, the reinforcement learning algorithm can be used in an online manner in which the system can learn and adapt in real time.

- Planning and Decision Making Process – This refers to the process of using knowledge obtained from learning phase to schedule and prepare for the next transmission. If multiple choices of actions are available, a transceiver must decide to choose the best strategy to achieve the target objectives. For example, when a CR sender schedules a transmission based on the frequency usage of a PU, a decision on the transmission power needs to be made to achieve the acceptable level of interference caused to the PUs.

- Action Process – This refers to the process of responding to the environment. Generally speaking, the action of a CRU is controlled by the planning and decision making process.

In terms of the spectrum access, the two major approaches used in CR technology are dynamic spectrum allocation and opportunistic spectrum access [11]. For dynamic spectrum allocation, information on spectrum occupation is used for channel allocation and planning on a long-term basis. On the other hand, with opportunistic spectrum access, instantaneous information of channel usages by PUs is observed and CR is granted access to utilize the spectrum holes. This can effectively increase the spectrum utilization on a short-term basis. It is believed that the opportunistic spectrum access provides more agility, but at the expense of higher computational complexity.

2.2 Introduction to IEEE 802.11 Networks

The IEEE 802.11 MAC [12] has become ubiquitous and gained widespread popularity as a layer-2 protocol for wireless local area networks (WLAN). This MAC layer is responsible for a structured channel access scheme and is implemented using a Distributed Coordination Function (DCF) based on the Carrier Sense Medium Access with Collision Avoidance (CSMA/CA) protocol. An alternative to the DCF is also provided in the form of a Point Coordination Function which is similar to a polling system for determining the users who have the right to transmit. In this section, we only describe the relevant details of DCF access method and refer the reader to [12] for other details on the IEEE 802.11 standard.

The CSMA/CA based MAC protocol of IEEE 802.11 is designed to reduce the collision due to multiple sources transmitting simultaneously on a shared channel. In a network employing the CSMA/CA MAC protocol, each node with a packet to transmit first senses the channel to ascertain whether it is in use. If the channel is sensed to be idle for an interval greater than the Distributed Inter-Frame Space (DIFS), the node proceeds with its transmission. In contrast, if the channel is sensed as busy, the node has to defer its transmission to prevent contention.

To do so, the node initializes its backoff timer with a randomly selected backoff interval and decrements this timer every time it senses the channel to be idle. The node is then allowed to transmit only when the backoff timer reaches zero. Since the backoff interval is chosen randomly, the probability that two or more stations will choose the same backoff value is very low. This can minimize the negative effects caused by channel contention.

Along with the Collision Avoidance, 802.11 MAC uses a positive acknowledgment

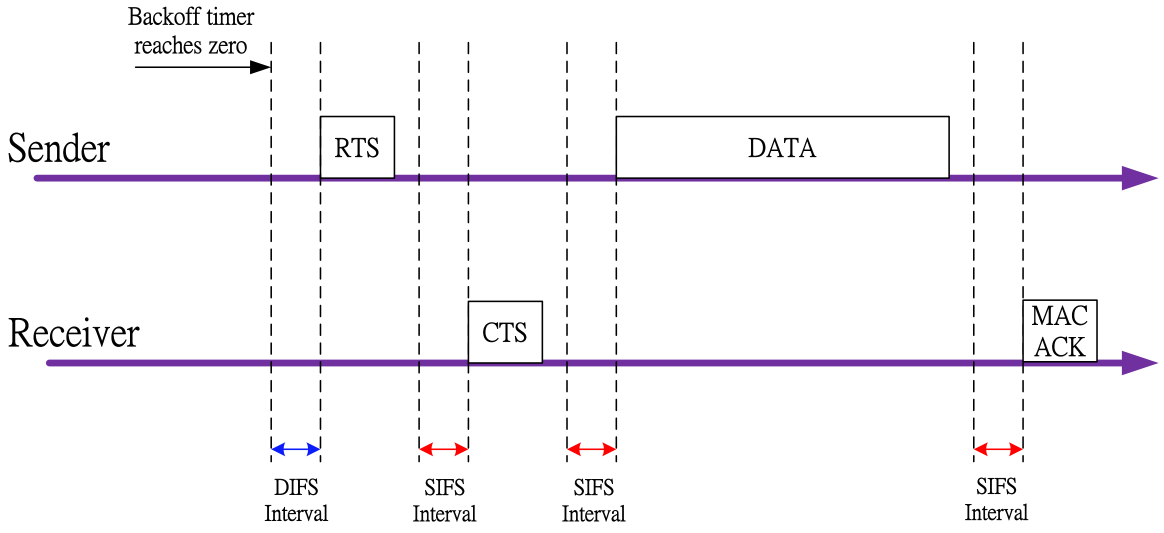


Figure 2.1: Basic Operation of the CSMA/CA Protocol

(ACK) scheme. All the packets received by a node implementing 802.11 MAC must be acknowledged by the receiving MAC. After receiving a packet the receiver waits for a brief period, called the Short Inter-Frame Space (SIFS), before it transmits the ACK.

There is another particular feature of WLANs, known as the “Hidden Terminal” (HT) problem, that the 802.11 MAC specification addresses. Two stations that are not within hearing distance of each other can lead to collisions at a third node which receives the transmission from both sources. To take care of this problem, 802.11 MAC uses a reservation based scheme. A station with a packet to transmit sends an Ready to Send (RTS) control packet to the receiver and the receiver responds with a Clear to Send (CTS) control packet if it is willing to accept the packet and is currently not busy. This RTS-CTS exchange, which also contains timing information about the length of the ensuing transaction, is detected by all nodes within hearing distance of either the sender or receiver or both. Therefore, all of these nodes defer their transmissions till the current transmission is completed to avoid packets collision.

The basic operation of the CSMA/CA based MAC protocol of IEEE 802.11 is shown in Fig. 2.1. It shows the exchange of various packets involved in each successful transmission and the spacing between these packets.

2.3 Cognitive Radio over IEEE 802.11 Networks

Recently the Cognitive Radio (CR) technology has been applied to the ubiquitous 802.11 networks [9][10][13][14][15]. To fast respond to the dynamic uses of the authorized frequency of an IEEE 802.11 network, all of these cited works use an opportunistic spectrum access approach to manage the channel accesses of CRUs without the control of a central unit (i.e., access point). [13][14][15] assume that CRUs can perfectly detect the end of a PU's packet transmission.

In [13], the authors propose a novel sense-transmit-wait strategy of CR MAC protocol by considering unsaturated 802.11 networks. In their work, it is assumed that each CRU is equipped with a SDR that can detect the transmission activities of PUs and maintain the cumulative distributed function of primary network's inter-packet arrival time. When a CRU detects that a channel is idle, it decides how long it can transmit such that the transmission time is as large as possible while the probability of collision satisfies the primary network's quality of service (QoS) requirement.

[14] proposes an opportunistic CR MAC protocol that is able to predict the interval of spectrum holes to maximize the allowed transmission time of CRUs. Using their protocol, a pair of CRU can exchange data only if they have an available (i.e., idle) channel in common. Therefore, a handshaking process along with channel selection mechanism (on the control channel) before data transmission (on the data channel) is covered in this protocol. Similarly, it is assumed that each CRU is equipped with SDR which can provides the PUs' information of channel utilization and average packet size in a specified time window periodically. Furthermore, each CRU also maintains a channel state table which records the system busy time. With all of these information, a CRU using the proposed opportunistic CR MAC protocol can estimate the primary utilization more accurately, thus can predict its average packet size for transmission in the next time slot.

Lastly, in [15], the proposed MAC protocol maintains a channel occupation ratio for each channel of the primary network (i.e., 802.11 network) to keep the communication quality of the primary system. The packet transmission control of the CRUs is then performed according to the channel occupancy ratios. This is done by first making a channel list in the order of the lowest channel occupancy ratio to the highest channel occupancy ratio, and then schedules the CRU's transmissions in a probabilistic manner based on the most recent channel occupancy ratio of the selected channel.

Using the above proposals, a CRU schedules data transmissions for a variable-length interval. The length of the interval is adjusted only at the end of the previous interval. However, it is commonly known that IEEE 802.11 networks are open systems where accurate traffic prediction required by these proposals cannot be easily achieved. To address this problem, TakChou Lou et al. [9] proposed a MAC protocol that allows CRUs to flexibly abort their data transmissions when they detect that PUs intend to access the authorized frequencies. In this design, a PU has opportunity to advertise its intention of claiming the channel through broadcasting RTS and CTS control messages during the periods which CRUs are silent.

Although the MAC protocol in [9] provides scheduling flexibility for CRUs to prevent their transmissions from interfering with PUs' transmission, its channel selection algorithm is inefficient for CRUs to quickly find an available channel to use. In addition, this protocol does not consider the impacts of HTs on network performances. To solve this problems, Shie-Yuan Wang et al. [10] further expanded TakChou Lou et al.'s work to address the above issues. They proposed 1) a Smart Channel Selection Scheme to reduce the time required by CRUs in finding an idle 802.11 channel for data transmission, and 2) an enhanced data transmission and channel evacuation approach to solve the HT problems. Through extensive simulations, although [10] can effectively improve the overall bandwidth utilization as compared to [9], it is found that TCP flows did not perform well under [10].

2.4 Related Work

In this section, we introduce the MAC protocol proposed by [9] and [10], respectively. These two previous works serve as the major related works of this dissertation.

2.4.1 Related Work I: On Synchronized Channel Sensing and Accessing for CR Users in IEEE 802.11 wireless Networks

In [9], TakChou Lou et al. considered the problem of how CRUs can opportunistically use the bandwidth of ubiquitous IEEE 802.11 networks. That is, PUs are users authorized to access the frequency used by the local IEEE 802.11 network with the DCF and CRUs are users unauthorized to access those frequencies. To solve this problem, in [9] they proposed a CR MAC protocol for CRUs to access the authorized frequencies in an opportunistic manner. In this protocol, an 802.11 channel is used as the control channel. CRUs should monitor control messages broadcast on the control channel whenever they are idle. The protocol defines three new control messages: 1) Request-To-Send-CR (REQCR); 2) Grant-To-Send-CR (GRANTCR) and 3) Suspend (SUS). When a CR sender intends to transmit a data packet, it should perform the REQCR/GRANTCR handshake with the CR receiver on the control channel.

The REQCR control message is an RTS control message plus the following extra fields: 1) the ID of the initial selected data channel (denoted as *InitCH*), and 2) increment-per-hop (denoted as *Inc*). *InitCH* denotes the initial data channel to which the pair of CR sender and receiver (denoted as CRU pair hereafter) will switch to perform data transmission, and it is randomly selected by the CR sender. *Inc* denotes a random number used for deriving the ID of the next data channel that the CRU pair will try to access. Given *InitCH*, *Inc* and the number of data channel (denoted as N_c), the next-hop channel is determined by the following equation:

$$CH(i) = (CH(i-1) + Inc)\ mod\ N_c,\ \ i \geq 2\ , \tag{2.1}$$

where $CH(1) = InitCH \leq N_c$.

For example, for an IEEE 802.11 network with eight data channels, given that *InitCH* = *2* and *Inc* = *3*, the derived channel hopping sequence (CHS) is (2, 5, 8, 3, 6, 1, 4, 7).

Upon receiving an REQCR sent by the CR sender, the CR receiver first uses Eq. 2.1 to derive the CHS. It then replies a GRANTCR to the CR sender, if it is ready to receive the data. Thereupon, it immediately switches itself to the *InitCH* indicated by the CHS. On the other hand, on receiving the REQCR, the CR sender should also immediately switch itself to the *InitCH* indicated by the CHS. Because the CRU pair use the same information to derive the CHS, their CHS on the data channel will be the same. Using this design, their channel hopping behaviors can be synchronized.

When switching themselves to a data channel, the CRU pair should first monitor the channel for a predefined channel sensing period to sense the channel status. If the channel is idle, the CRU pair will carry out the original RTS/CTS/DATA/ACK transmission sequence. Otherwise, they synchronously switch themselves to the next data channel indicated by the CHS. Such a channel hopping behavior repeats until they can find an

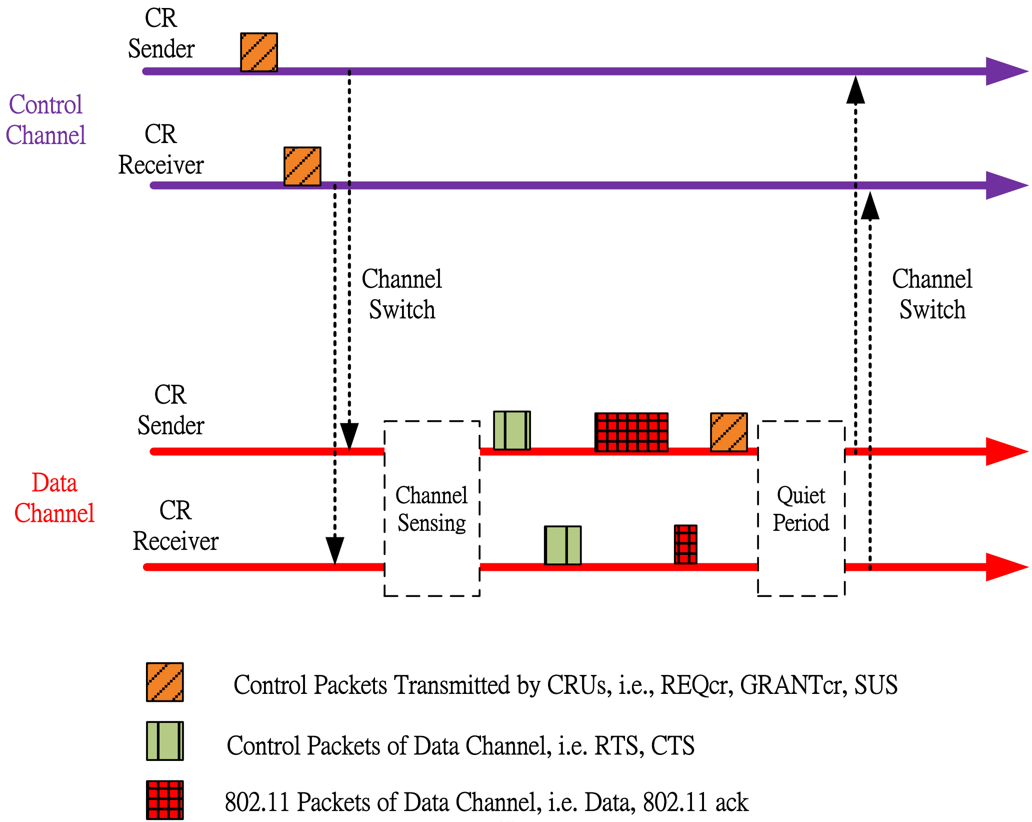


Figure 2.2: Illustration of Packets Transmission of [9] when Txop = 1

idle data channel to transmit a data packet. The CHS should be wrapped to the *InitCH* if all of the data channels have been sensed.

When successfully accessing a data channel (i.e., succeeding in exchanging the RTS and CTS packets), the CRU pair are allowed to transmit *Txop* frames, where *Txop* denotes the number of Transmission Opportunities. The CR sender should transmit a SUS control message to the CR receiver after successfully transmitting a data frame (i.e., upon receiving the corresponding ACK frame). After transmitting the SUS control message, the CR sender should keep silent on this data channel for a predefined queit period (QP). This QP provides opportunities for PUs to broadcast RTS and CTS control packets on the data channel to reclaim the use of this data channel. Therefore, during the QP, the CRU pair will proactively evacuate from the data channel and return to the control channel upon overhearing RTS or CTS control messages.

If the *Txop* is set to one, the CRU pair should should return to the control channel after the QP. The packets transmission between the CRU pair when *Txop* is being set to one is illustrated in Fig. 2.2.

If the *Txop* is set to any value larger than one, the CRU pair remain to stay on the

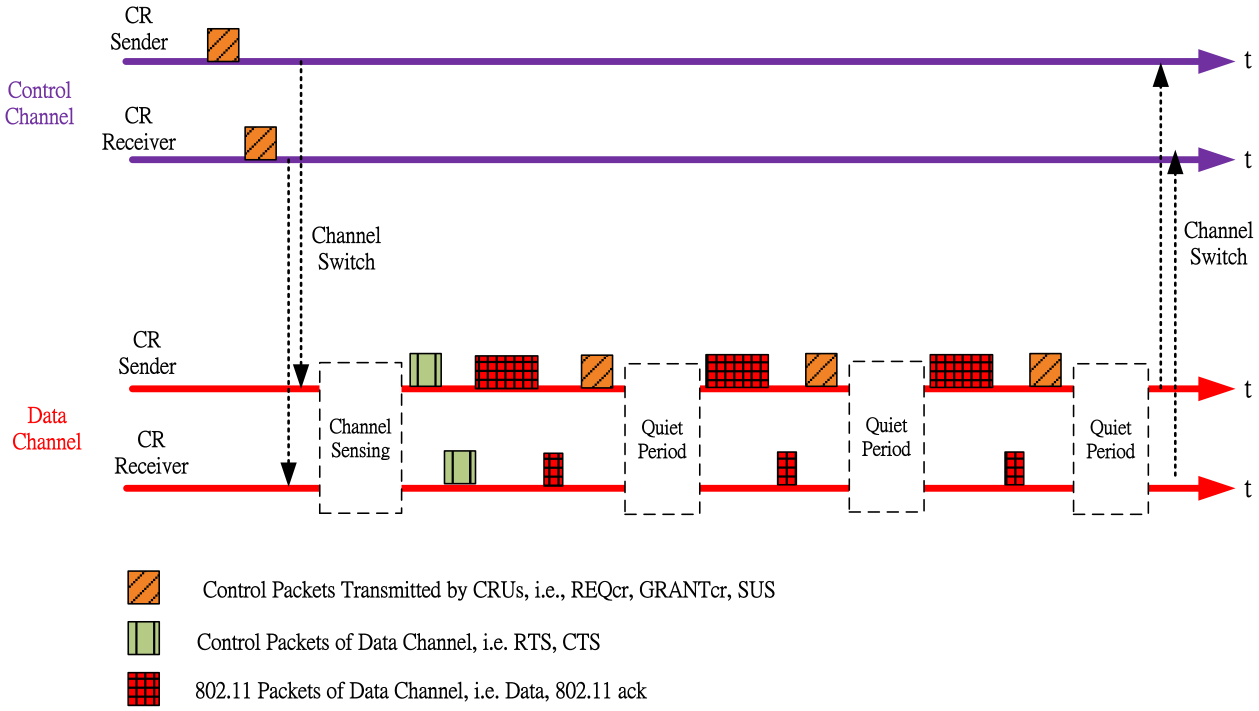


Figure 2.3: Illustration of Packets Transmission of [9] when Txop = 3

channel after the QP if the data channel stays idle throughout the QP. The CRU pair will regard that the channel remains unused by the PUs. As such, they continue to borrow the bandwidth of the data channel and immediately start a new DATA/ACK/SUS transmissions as described above. After *Txop* DATA/ACK/SUS transmissions, the CRU pair should immediately return to the control channel. Fig. 2.3 depicts the packets transmission when the *Txop* is being set to three.

Next, we illustrate the packets transmission when a data channel hopping is required. Assumed that for an IEEE 802.11 network with eight data channels, given that $InitCH = 2$ and $Inc = 3$, the derived CHS (2, 5, 8, 3, 6, 1, 4, 7). After a successful REQcr/GRANTcr handshaking process on the control channel, both the CRU pair will switch to Channel 2 as indicated in the CHS for subsequent data transmission and data reception. We further assume that during the channel sensing period of monitoring the Channel 2, PU's activity is sensed by both the CRUs. In this case, upon the termination of channel sensing period, the CRUs switch synchronously to next data channel as indicated in the CHS, i.e., Channel 5, and repeat the channel sensing procedure on Channel 5. Fig. 2.4 depicts the above scenario where the CRU pair manage to find an idle data channel to transmit data after one time of data channel hopping.

The CR MAC protocol proposed in [9] has two drawbacks. First, it does not assess

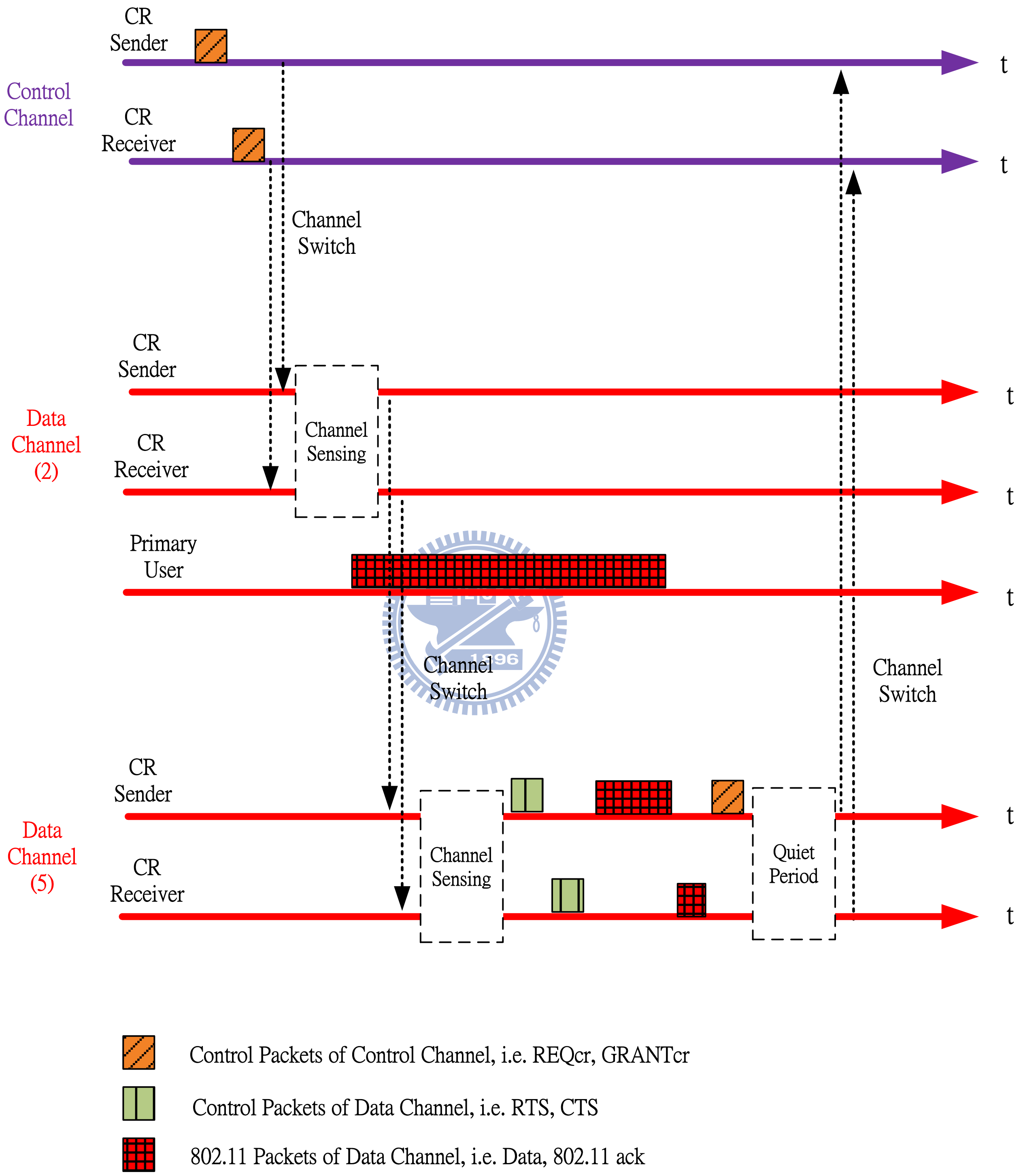


Figure 2.4: Illustration of data channel switching when a data channel is sensed busy

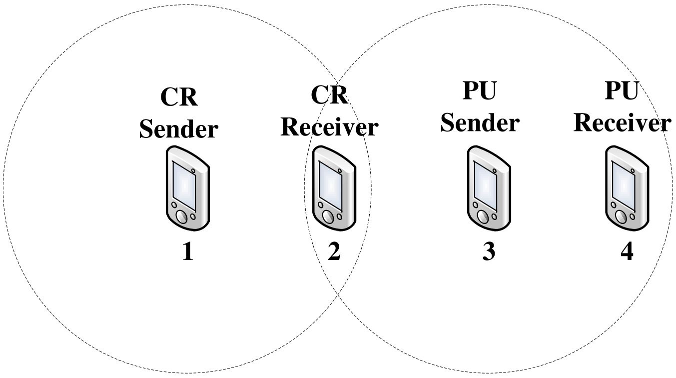


Figure 2.5: Illustration of Hidden Terminals

the statuses of data channels before deciding the CHS of a CRU pair. Therefore, the CRU pair may switch themselves to the *InitCH* and find that it is occupied by PUs. In this condition, they need to hop to the next data channel and sense the PUs' behavior on the data channel again. Assuming that the primary network has only one unused data channel at this time point, using this design the CRUs may need to switch N_c-1 times to find out the unused channel for data transmission in the worst case. Thus, this CR MAC protocol is inefficient in finding an available data channel.

Second, this proposed CR MAC protocol does not consider the effects of HTs which commonly exist in wireless networks. Fig. 2.5 illustrates an example network with the existence of HTs, where nodes 1, 2, 3, and 4 are CR sender, CR receiver, PU sender and PU receiver, respectively, and nodes 1 and 3 are HTs to each other. In this example, the RTS control message sent by the PU sender for claiming the use of the data channel cannot be overheard by the CR sender. Therefore, after the QP, the CR sender will consider that the data channel remains idle and continues its data transmissions. However, this wrong decision will cause the following data frame transmitted by the CR sender to collide with those transmitted by the PU sender at the CR receiver. As a result, the flow goodputs at the CR receiver will be greatly decreased when HTs exist.

To solve these drawbacks, Shie-Yuan Wang et al. [10] proposed an enhanced CR MAC protocol as elaborated in the following section.

2.4.2 Related Work II: Enhanced MAC Protocol for Cognitive Radios over IEEE 802.11 Networks

As a remedy to the problems as identified in Section 2.4.1, Shie-Yuan Wang et al. [10] proposed two enhancements to [9]'s CR MAC protocol: 1) a Smart Channel Selection Scheme, and 2) an enhanced data transmission and channel evacuation approach. The details of the enhancements are elaborated in the following sections.

I. Smart Channel Selection Scheme (SCSS)

In [10], a CRU pair do not use Eq. 2.1 to derive the CHS on the data channels. Instead, they use a handshake protocol to negotiate the order of CHS. To do so, each CRU should maintain a 32-bit Availability Record (AR) for each data channel to record its historic availability sensing results. A CRU updates the AR of a data channel each time when it performs the availability checking process and a 100 microseconds fast channel sensing process on this data channel. Each bit of an AR records whether the data channel is idle or not upon the execution of these two processes. The value of a bit being one means that the channel is idle and the value of a bit being zero means that the channel is busy. Before updating an AR, the CRU should right-shift the value of the AR by one bit. It then updates the most significant bit (MSB) of the AR. This means that the MSB of an AR records the latest status of a data channel.

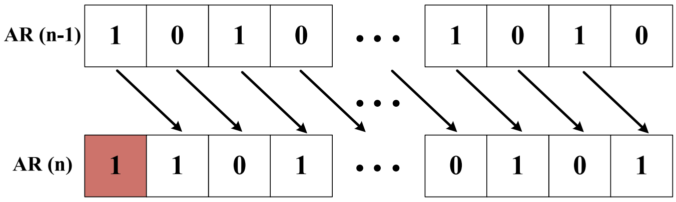
The availability of a data channel is determined based on the following rules:

- Overhearing an 802.11 ack packet indicates that a data transmission on this data channel is just completed. In this case, the status of this data channel is considered as **IDLE**.

- Overhearing an RTS or CTS control packet indicates that a data transmission is about to begin. In this case, CRUs should update the Network Allocation Vector (NAV) for this data channel based on the overheard RTS/CTS frame and consider the status of this data channel as **BUSY**.

- Overhearing a DATA packet indicates that a data transmission is being performed. In this case, the status of this data channel is considered as **BUSY**.

- Not overhearing any MAC-layer frames indicates that the data channel is currently



AI (n-1) = 1x31 + 0x30 + 1x29 +…+ 0x2 +1x1 + 0x0 = 256

AI (n) = 1x31 + 1x30 + 0x29 +…+ 1x2 +0x1 + 1x0 = 271

Figure 2.6: Calculation of the Availability Index (AI)

unused and available for CRUs to transmit data. In this case, the status of this data channel is considered as **IDLE**.

When selecting candidate data channels for data transmission, the CR sender should first compute the Availability Index (AI) for each data channel using the following equation:

$$AI = \sum_{i=0}^{31} b(i) * pos(i), \tag{2.2}$$

where $b(i)$ denotes the value of i-th bit and $pos(i)$ denotes the position of the bit from the least significant bit to MSB. Fig. 2.6 shows two examples of calculating the AI value. The AI value reflects the idle probability of a data channel. A data channel with the largest AI value is the most preferable channel to be used for exchanging data because 1) it is likely to be idle in the near future or 2) it has remained unused at most time.

The CR sender then sends an REQCR message to the CR receiver. The REQCR message is the original REQCR message in [9] with the *InitCH* and *Inc* being replaced by a 16-bit preferable channel (*CH_pr*) field to indicate the preferable channels for data transmission. Each bit in the *CH_pr* indicates whether a data channel is selected as a candidate by the CR sender. As explained above, the channel selection is based on the AI value of each data channel.

Upon receiving an REQCR message, the CR receiver should perform a 100 microseconds fast channel sensing process to check the availability of each selected data channel. In the fast channel sensing process, the CR receiver switches itself to each selected data channel, stays on the data channel for 100 microseconds, and senses the availability status of that data channel. 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 CHS among the selected data channels. Finally, the CR receiver should broadcast a GRANTCR message

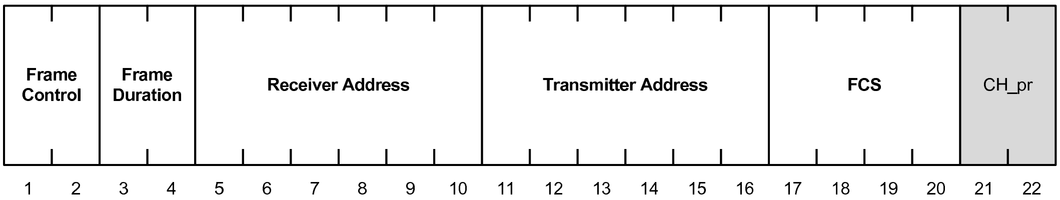


Figure 2.7: Frame Format of the control packet REQCR

Figure 2.8: Frame Format of the control packet GRANTCR

to acknowledge the CR sender. The GRANTCR message is modified such that it contains a 32-bit channel hopping order (*CH_HO*) field, to store the CHS determined by the CR receiver. Fig. 2.7 and Fig. 2.8 depict the frame format of REQCR and GRANTCR control packets, respectively.

The rationale behind this design can be explained using Fig. 2.5. The transmission activities of PUs (e.g., node 3 in Fig. 2.5) behind the CR receiver may not be detected by the CR sender (e.g., node 1 in Fig. 2.5), due to the limited carrier-sense range of its radio. That is, these PUs and the CR sender are HTs to each other. If they simultaneously transmit data on the same data channel, their data will be collided at the CR receiver, greatly decreasing the goodputs obtained by the CR receiver. Using [9], the CR sender cannot know the transmission activity of such a PU until it is on the data channel and has not received the CTS upon timeout.

In contrast, using [10] the CR receiver can quickly notify the CR sender of the detected PU's transmission activities on a data channel by giving this channel a lower preference in (or excluding it from) the *CH_HO* field. Thus, [10] can effectively reduce the time required to find an available data channel to transmit data.

II. Enhanced Data Transmission and Channel Evacuation

[9] does not consider the existence of HTs. After exchanging the RTS/CTS control messages, the CR sender is allowed to transmit a maximum of *Txop* data frames before

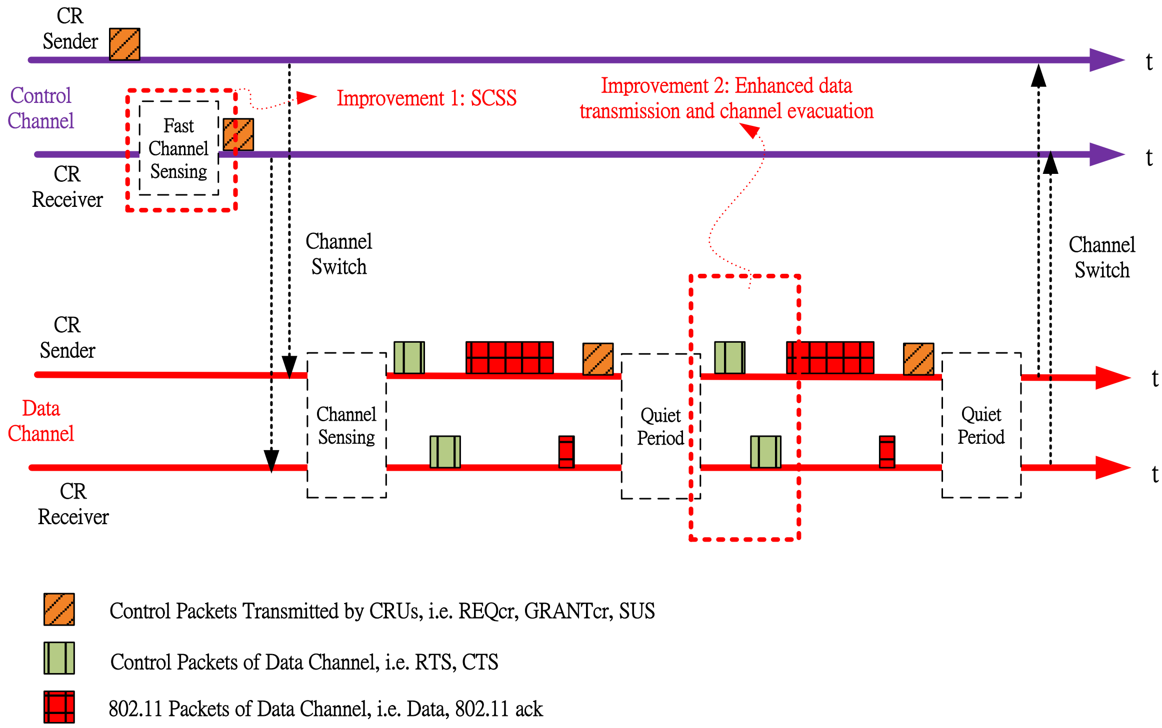


Figure 2.9: Illustration of Message Exchanges on Data Channel of [10] when $Txop = 2$

returning to the control channel. In between the *Txop* data frame transmission, QP is introduced for PUs to reclaim the channel. However, during this QP, if the PUs are hidden from the CR sender, the CR sender will not be aware of them even if the PUs have exchanged the RTS/CTS messages for channel reservation. After the QP, the CR sender will continue to transmit data frame to the CR receiver. As a result, the data frame will be collided at the CR receiver and decreases the network goodputs.

To address this problem, in [10] the CRU pair are required to complete an RTS/CTS handshake procedure before every data frame transmission on a data channel. Failure to finish the RTS/CTS handshake procedure indicates that the data channel is currently used. In this condition, the CRUs should proactively evacuate from the data channel and return to the control channel. Although performing the RTS/CTS handshake procedure prior to each data transmission will add some bandwidth overhead, it effectively prevents the data transmissions of CRUs and those of PUs from interfering with each other, thus significantly increasing the goodputs of the network when HTs are present. Fig. 2.9 illustrates the enhanced CR MAC protocol as proposed by [10].

2.5 Problem Statement

Through both the proposed enhancements by [10], it is proven qualitatively through simulations that the overall bandwidth utilization of [10] outperforms [9]. However, when the TCP performance is compared against the UDP performance, it is found that the TCP flows obtained a low aggregate throughputs.

After a detailed study and investigation, we revealed that the TCP flows remained to perform poorer than that in conventional 802.11 networks even when PUs are absent in the CRN. A possible explanation for this finding is as follow: In [10], every successful bandwidth negotiation procedure (i.e., exchange of REQCR and GRANTCR control packets) only reserves a one-way bandwidth for the CR sender to transmit a maximum of *Txop* packets. This protocol works fine when the considered traffic type is uni-directional, i.e., one-way UDP flow. However, in the presence of bi-directional traffic, such as TCP flows with reverse ACKs, the performance of this protocol was degraded due to its inadequate support to reserve bi-directional bandwidth.

Since TCP is the most widely used transport-layer protocol over the internet, it is imperative to design a CR-MAC protocol which can well support TCP flows. Therefore, in the following chapter, we first propose a basic scheme of bi-directional CR MAC protocol to overcome the limitation of [10] in supporting traffic flows with bi-directional packets, particularly the TCP flows. Next, we propose an advanced scheme of bi-directional CR MAC protocol which can dynamically reserve a short-term bandwidth for the CRU pair to carry asymmetry traffic flows. For brevity, in the following chapters, we denote the MAC protocol in [10] that favors the uni-directional traffic as “Uni-MAC”, the proposed basic scheme of bi-directional CR MAC protocol as “BBi-MAC” and the proposed advanced scheme of bi-directional CR MAC protocol as “ABi-MAC”.

Chapter 3

Proposed Bi-directional CR MAC Protocol

3.1 Motivation

In Section 2, we have explained the CR MAC protocol which was originally proposed by by [9] and later enhanced by [10]. Although both of these related works have dramatically improved the spectrum efficiency, their studies fail to address the performance degradation of their protocols in delivering internet applications with the characteristics of bi-directional packets transmission and variable packet sizes. The most common examples of such internet applications include FTP flows which are transported through TCP and bi-directional UDP flows to carry conversational VoIP calls.

In addition, the main limitation of their protocols is that the short-term bandwidth allocation for the CRUs has to be done statically. Another way of saying, the bandwidth allocation for these previous works depends on a pre-defined network parameter, denoted as Transmission Opportunity (*Txop*), such that during every successful channel access, a CRU is permitted to transmit only a maximum of *Txop* packets. This, however, limits the protocol performance as the network traffic varies with time and therefore a fixed bandwidth allocation may not be able to meet the instantaneous traffic demand of CRUs optimally.

Therefore, we propose a bi-directional CR MAC protocol which is an extension of the CR MAC protocol proposed by [9] and [10]. In this proposal, we divide the development into two phases. In phase one, we focus on the basic scheme of bi-directional CR MAC protocol (BBi-MAC) to better support TCP flows and bi-directional UDP flows. In phase two, we work on an advanced scheme of bi-directional CR MAC protocol (ABi-MAC) with short-term and smart dynamic bandwidth allocation.

3.2 Proposed Basic Scheme of Bi-directional CR MAC Protocol (BBi-MAC)

3.2.1 Motivation and Problem Elaboration

The main idea of the proposed BBi-MAC is to better support traffic flows with bi-directional packets transmission, particularly the TCP flows and bi-directional UDP flows such as conversational VoIP calls. To this end, we define *two-way* or *bi-directional* traffic as the traffic pattern resulting from: 1) one or more TCP connections transferring data between a CRU pair, 2) two or more UDP connections transferring data in opposite directions between a CRU pair, and 3) a combination of at least a TCP flow and at least an UDP flow between a CRU pair.

It is commonly known that for TCP flows, a congestion control mechanism is employed. The essence of this congestion control mechanism is the observation that data packets arrive at the receiving host at the rate that the bottleneck link will support. 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 efficiency. 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.

In addition, the throughput of a TCP flow could also be affected by its round-trip time (RTT). The RTT is the time between when a TCP data is being put on the wire and when its corresponding ACK is received. In TCP connections, every TCP receivers keeps a Receive Window (RWIN), which refers to the amount of data that the TCP receiver can accept without acknowledging the TCP sender. On the other hand, a TCP sender can transmit data up to the window size before waiting for the TCP ACKs. The TCP throughput T limitation caused by the TCP window size can then be calculated as follows:

$$T \leq \frac{RWIN}{RTT}. \tag{3.1}$$

In most of the system, the RWIN has a default size of 64KBytes. Therefore, it is obvious that when the RWIN is fixed, the variation of RTT can affect the TCP throughput. A smaller RTT can increase the maximum achievable throughput while a large RTT can decrease the TCP throughput.

From detailed observation, we have identified the occurrence of *ack-compression* effect in the Uni-MAC in transmitting TCP flows. Also, it is found that the RTT time of a TCP packet becomes longer due to the protocol design. These observations are further elaborated as follow.

Consider a CRN with only a single pair of CRUs, transmitting a TCP flow from CRU A to CRU B, and the *Txop* being set to two. As explained earlier, TCP's *self-clocking* depends on the arrival of TCP ACKs at the same spacing with which the receiver generated them. This desire same spacing of TCP ACKs, however, is broken in the Uni-MAC as proposed by [10].

With the Uni-MAC, when the TCP data are transmitted, the TCP ACKs could not be replied immediately due to only one-way bandwidth reservation by the protocol. Instead, the TCP sender first sends *Txop* TCP data packets consecutively. After that, both CRU A and CRU B return to the control channel. At this time point, since both the CRUs have packets in queue to be transmitted (TCP data for CRU A and TCP ACK for CRU B), both of them have to compete for the next chance of bandwidth negotiation, i.e., transmitting an REQCR.

In the Uni-MAC, to ensure that the control channel is accessed fairly by the CRUs, every CRU has to defer for a random waiting duration (RWD), which is a factor of SIFS, before it is allowed to transmit an REQCR. Such a design gives every CRUs an equal chance to be a CRU sender. In the best case, the CRU B (denoted as TCP receiver hereafter) may defer for a shorter RWD as compared to CRU A (denoted as TCP sender hereafter). Therefore, it will get a chance to transmit an REQCR before the TCP sender and switch its role to be a CR sender. After a successful REQCR/GRANTCR handshake at the control channel and obtaining the channel access at the pre-negotiated data channel, the TCP receiver can transmit its accumulated TCP ACKs after some standard procedures.

The resulting TCP transmission sequence of the best case as elaborated above is shown in Fig. 3.1. However, one can see that such data transmission sequence is not favorable because it breaks the TCP self-clocking nature due to the *ack compression* effect. When this happens due to the deficiency of the protocol design, the TCP sender has to tune the interval it sends TCP data and thus degrades the TCP transmission rate.

In an unpreferable situation, the TCP sender may get a shorter RWD continuously as compared to the TCP receiver as shown in Fig. 3.2. Worst of all, this might happen repeatedly until the TCP sender has used up its TCP window size and thus can only transmit the next TCP data after it receives a TCP ACK. Therefore, the TCP sender will no longer compete for the next chance of bandwidth negotiation and provides the TCP receiver a full opportunity to transmit an REQCR control packet after a RWD. As a consequence, the TCP receiver could transmit the accumulated TCP ACKs only some time later and this results in a more severe TCP ACKs compression and degrades the TCP throughput drastically.

The TCP performance degradation problem will be further deteriorated if the considered network consists of some other CRUs and PUs which are also competing for the bandwidth. In this situation, the TCP ACKs compression problem will get even more severe because (i) The TCP sender has to compete with others CRUs for the bandwidth negotiation on the control channel. Therefore, its chances of immediately transmitting an REQCR becomes slimmer; and (ii) Even if the TCP sender has completed the handshaking procedure with the TCP receiver and both have switched to the data channel, it might not be able to get an immediate chance of packet transmission if PU's activities are sensed on that channel. Combination of these two factors can further prolongs the time between a TCP data is sent and the corresponding TCP ACK is received, i.e., the RTT.

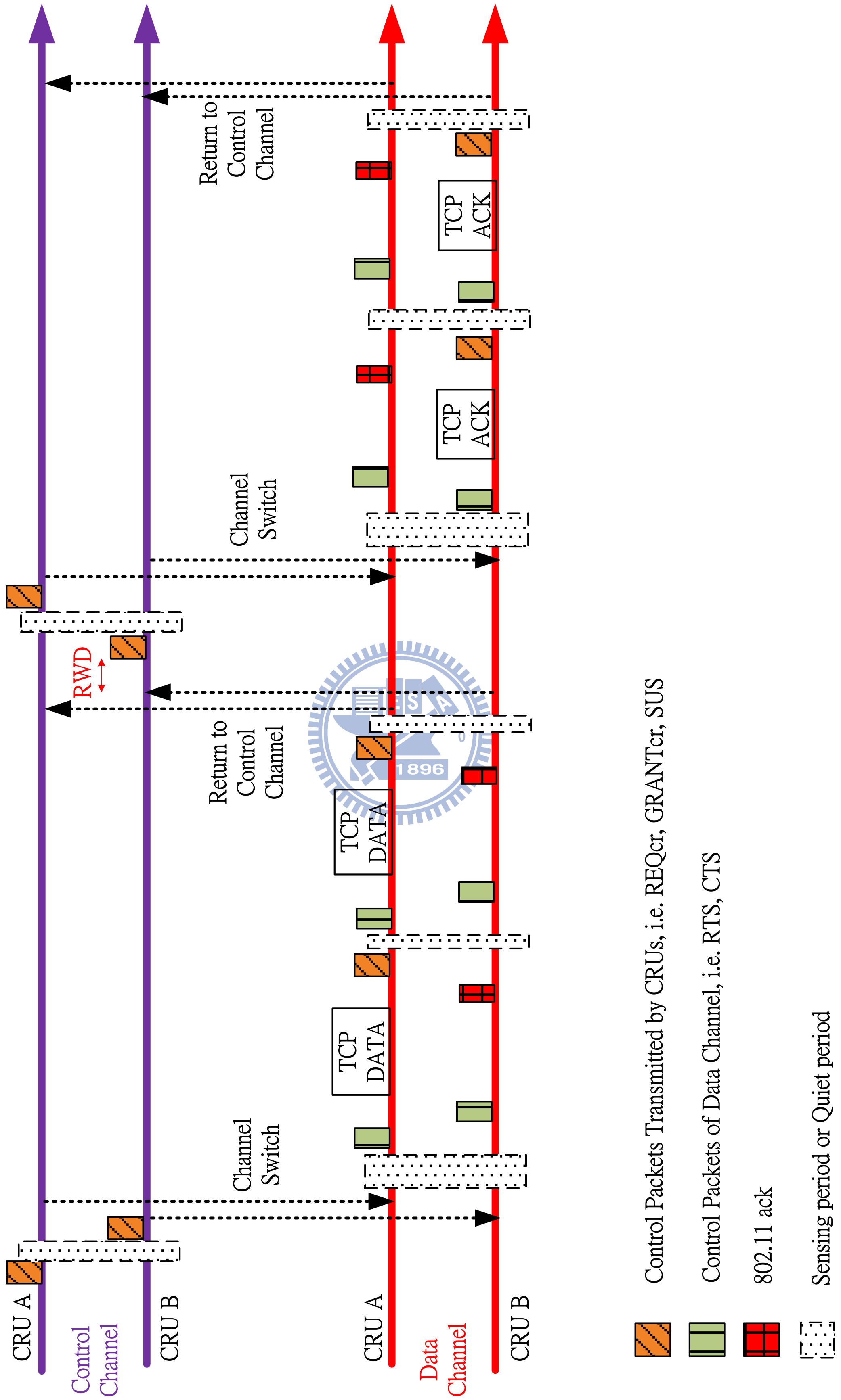


Figure 3.1: Illustration of the best case of the TCP packets exchanges when *Txop = 2*

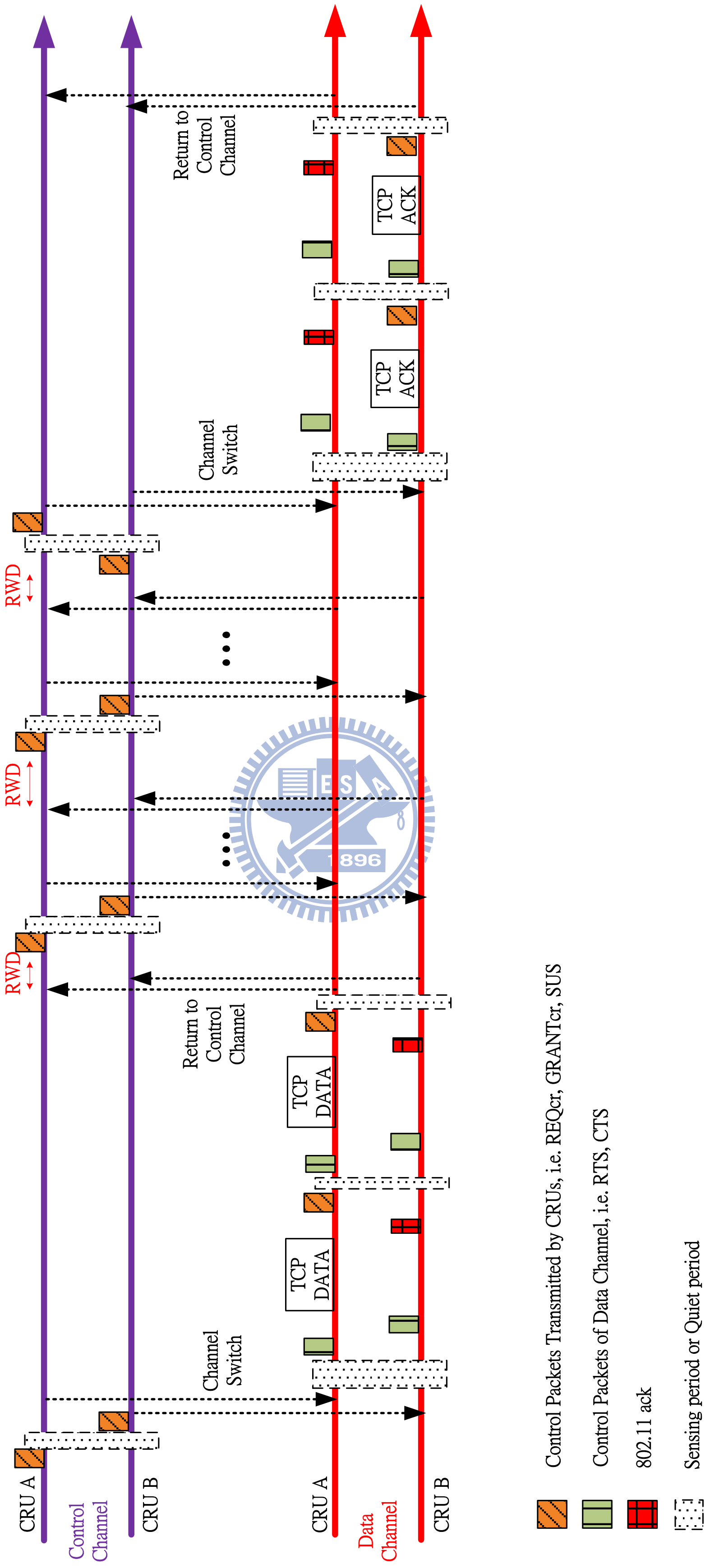


Figure 3.2: Illustration of a bad case of the TCP packets exchanges when *Txop = 2*

3.2.2 Design of the BBi-MAC

To address the problem as elaborated in Section 3.2.1, the BBi-MAC is designed to reserve bandwidth according to the anticipated traffic type. In BBi-MAC, the control packets, i.e., REQCR and GRANTCR are extended with additional 2-bit field. This extended field is named as *Reservation Type (RT)*, which serves as an indicator for a CRU pair to reserve the bandwidth either uni-directionally or bi-directionally. Before sending an REQCR or a GRANTCR, a CRU always scans its head of queue (HOQ) to check whether there is any packet to be transmitted. If there is, the CRU first identifies the packet type and then fills in the RT field according to the principles as explained below.

As a CR sender, before transmitting an REQCR, if an UDP packet is identified as the HOQ, the CR sender first assumes a uni-directional traffic from itself to the CR receiver. In this case, it fills the RT of the REQCR (denoted as RT_{req} hereafter) with a value of *00* to indicate a single-way bandwidth reservation. In contrast, if the HOQ is a TCP packet, the CR sender would automatically assume a bi-directional traffic.

Such an assumption is valid because TCP is a reliable stream delivery service that guarantees delivery of a data stream through a technique known as positive acknowledgment with retransmission. This fundamental technique requires the receiver to respond with a TCP ACK as it receives the data. Therefore, if a TCP flow is initiated by the CR sender, it is reasonable to anticipate a bi-directional traffic. In this case, it fills the RT_{req} with a value of *01* to indicate a two-way bandwidth reservation. Once the RT_{req} is determined, the REQCR is transmitted over the control channel after the RWD.

As a CR receiver, upon receiving the REQCR, it first checks if it has any queued packets destined to the REQCR sender. If the queue is non-empty, apparently a two-way bandwidth reservation is necessary so that both the nodes have the opportunity to transmit the queued packets once they successfully gain access to a data channel. In this case, the RT field of the GRANTCR (denoted as RT_{grt} hereafter) is filled with a value of *10* which indicates that the CR receiver is demanding a reverse bandwidth reservation. In contrast, if the queue is empty, the CR receiver simply copies the value of RT_{req} to RT_{grt}. Finally, a GRANTCR is sent to the CR sender after the RT_{grt} and CHS are determined.

Upon receiving a GRANTCR, the CR sender performs a final check on the RT_{grt} value to determine the bandwidth reservation type. In general, a single-way bandwidth reservation should be performed if the RT_{grt} is filled with *00*. Alternatively, a two-bandwidth reservation would be performed by the CR sender if the RT_{grt} is filled with either *10* or *01*.

On the data channel, the transaction interval (TI) to be filled in a transmitted RTS control packet is decided based on the bandwidth reservation type. In the basic scheme, we only consider a simple network case with either a TCP flow or bi-directional UDP flows with a uniform packet size between a CRU pair. With a single-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 RT_{grt} value. If the RT_{grt} is filled with *10*, a TCP flow is anticipated. Therefore, the TI will be set to a duration which is adequate for transmitting a TCP data, a TCP ACK and two 802.11 ack packets. Alternatively, if the RT_{grt} 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.

In addition to differentiate the bandwidth reservation types through the exchanges of CR-related control packets, we also eliminate the transmission of SUS control packets in the BBi-MAC. Since the *Txop* is a fixed network parameter, the CRUs should always keep track of the number of transmitted or received packets whenever they are on the data channel. After every successful 802.11 ack packet reception (transmission), the CR sender (CR receiver) should automatically enter the QP if the recorded number of transmitted (received) packets is less than *Txop*. If the number of transmitted (received) packets is equal to the *Txop*, the CR sender (CR receiver) should return to the control channel immediately after it receives (transmits) an 802.11 ack packet. With this modification, the BBi-MAC can further reduce the overhead caused by transmitting the SUS control packets on the data channel.

Fig. 3.3 illustrates the packets exchanges on a data channel of Uni-MAC and BBi-MAC when the *Txop* is being set to two. In this example, we only consider a simple network topology with a CRU pair such that the CR sender is sending a TCP flow to the CR receiver. Fig. 3.3(a) depicts the packets exchanges of the CRU pair using the Uni-MAC under the best case scenario as elaborated in Section 3.2.1. Using the Uni-MAC, one can see that the TCP ACK cannot be replied immediately due to the one-way bandwidth reservation. Instead, a TCP receiver could only reply the accumulated TCP ACKs only after going through a sequence of the following actions:

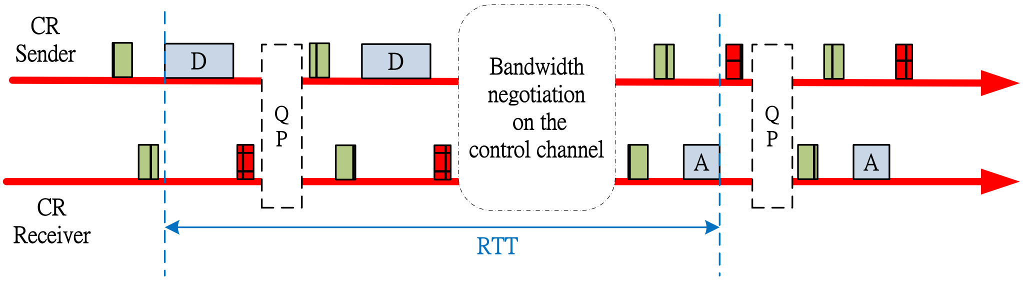
1. The TCP sender has transmitted *Txop* TCP data.
2. The CRU pair return to the control channel.
3. The TCP receiver has deferred for a RWD time period.
4. The TCP receiver has transmitted an REQCR and completed the handshaking procedure with the TCP sender.
5. The CRU pair switch to the data channel as pre-negotiated.
6. The data channel is sensed idle by the CRU pair.
7. The TCP receiver has completed the RTS-CTS handshake with the TCP sender.
8. Finally, the TCP receiver can transmit *Txop* TCP ACK.

Obviously, such a long sequence of inevitable actions in the Uni-MAC can cause an undesirable delay to the TCP ACKs and thus prolong the average RTT of a TCP flow. From the figure, one can see that even in the best case scenario, the RTT of a TCP data is several times larger than that using the BBi-MAC.

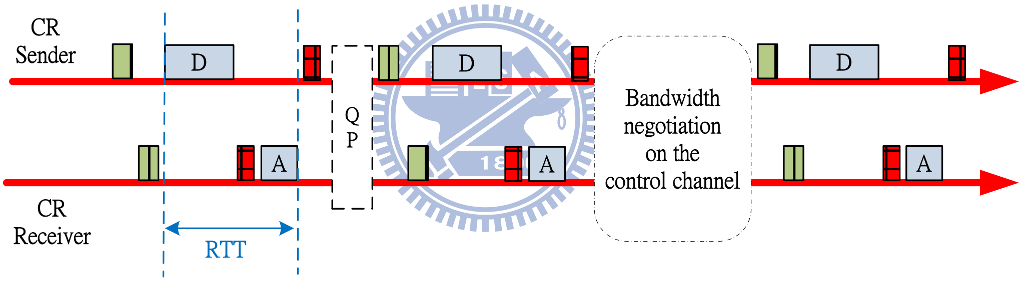
In contrast, due to the design of two-way bandwidth reservation when a TCP flow is detected, Fig. 3.3(b) shows that the BBi-MAC has effectively shorten the RTT of a TCP data. Therefore, it is believed that through the proposed BBi-MAC, better support can be offered to both uni-directional and bi-directional traffic flows in a CRN, particularly the TCP flows.

3.3 Proposed Advance Scheme of Bi-directional CR MAC Protocol (ABi-MAC)

The main idea of the proposed ABi-MAC is to allocate bandwidth to the CRUs in a more dynamic manner. To this end, we propose a Dynamic Bandwidth Allocation (DBA) algorithm, which can reserve bandwidth to the CRU pair dynamically based on the queue status. With this proposed design, the ABi-MAC not only can support a more



a) Uni-MAC (Txop = 2)



b) BBi-MAC (Txop = 2)

🟩	Control Packets of Data Channel, i.e. RTS, CTS	D	TCP DATA
🟥	802.11 ack	A	TCP ACK

Figure 3.3: Illustration of Packets Exchanges on Data Channel of a)Uni-MAC and b)BBi-MAC, when $Txop = 2$

complicated network cases, i.e., network with asymmetry two-way traffic with variable packet sizes, but also offers a better protection to both the CRUs and PUs. Also, the ABi-MAC can further utilize the “spectrum holes” in a more effective way and provide a better QoS to the CRUs in a CRN.

3.3.1 Motivation and Problem Elaboration

In Uni-MAC and BBi-MAC as elaborated in Section 2.4.2 and Section 3.2, *Txop* is adopted as a fixed network parameter. In Uni-MAC, *Txop* is defined as the number of packets that could be transmitted by a CR sender once it gains access to the data channel. In BBi-MAC, the definition of *Txop* is slightly ambiguous. Depending on the types of bandwidth reservation, the *Txop* is defined as:

- Uni-directional bandwidth reservation: The number of packets that could be transmitted by a CR sender once it gains access to the data channel.
- Bi-directional bandwidth reservation: The number of *transmission turns* for the CRU pair. For example, if *Txop* is set to a value of two, logically after every RTS/CTS handshake, CR sender and CR receiver each is given a *transmission turn* for sending a packet. However, if the queue of the CR receiver is empty during its turn, it could simply bypass its turn of packet transmission. When the timers expire, both the nodes will enter the QP. This shall be continued until they finish their *Txop*-th transmission turn.

Although the BBi-MAC gives flexibility for different bandwidth reservation types, the fixed value of *Txop* limits the protocol’s performances. Generally, a small value of *Txop* should be adopted when the network is busy and a large value of *Txop* is preferable when a network stays idle most of the time. However, since the network traffic varies across time, it is impossible for a pre-defined *Txop* to optimize the network throughput all the time.

In particular, a fixed value of *Txop* could affect the protocol’s performances in two ways. Firstly, when the *Txop* is set to a small value, i.e., one, it is considered as an appropriate setting when the network is in a busy condition. This is because with such setting, every CRU is given only one chance of packet transmission and after the packet transmission it has to return to the control channel. Therefore, the PUs are given more chances to use the channel, especially when the CRUs are away from the data channel.

However, when the network is in an idle status, the protocol can result in a very high overhead if the $Txop = 1$ is set to a value of one. This high overhead is caused by the time wasted when (i) the CRU pair return to the control channel and repeat the bandwidth negotiation procedure by exchanging CR control messages and (ii) the CRU pair switch to the data channel, sense the channel for a pre-determined sensing period, and exchange the RTS and CTS control packets before the data transmission can formally take place. Therefore, this high overhead could severely limit the protocol's performance in terms of the achievable maximum throughput.

Secondly, when the *Txop* is set to a large value, i.e., four, it is a suitable setting when the network is in an idle condition. With such setting, a CRU pair spend a longer time on the data channel for possible packets transmission. Furthermore, because of this protocol design, the PUs are still given chances to claim the channel when the CRUs enter the QP. However, when bi-directional bandwidth reservation is considered, overhead could be incurred when either one side of the CRU has no packet to send on their transmission turn.

Fig. 3.4 illustrates the message exchanges of a CRU pair on a data channel. They are operating using the BBi-MAC with bi-directional bandwidth reservation and the *Txop* is being set to a value of three. We consider a network scenario such that the CRU pair are transmitting asymmetry traffics (i.e., packets with variable sizes) over the CRN. One can see that initially both the CRUs have packets in queue to be transmitted. However, after the first transmission turn, the queue of the CR receiver becomes empty and remains empty even until the beginning of its second transmission turn. In this case, both the CRUs have to wait for a reasonable duration before their timers expire and both synchronously enter the QP. Obviously, the time spent waiting for the timers to expire has resulted in inevitable overheads and this negative impact might deteriorate if the CR sender remains to have an empty queue until its *Txop-th* transmission turn.

Such an undesirable phenomena not only increases the protocol overhead, but could also severely affects the PUs' performances. The reason is explained as follow: The BBi-MAC is designed to carry either one TCP flow between a CRU pair or two UDP flows of opposite directions and with uniform size between a CRU pair. As such, in BBi-MAC the

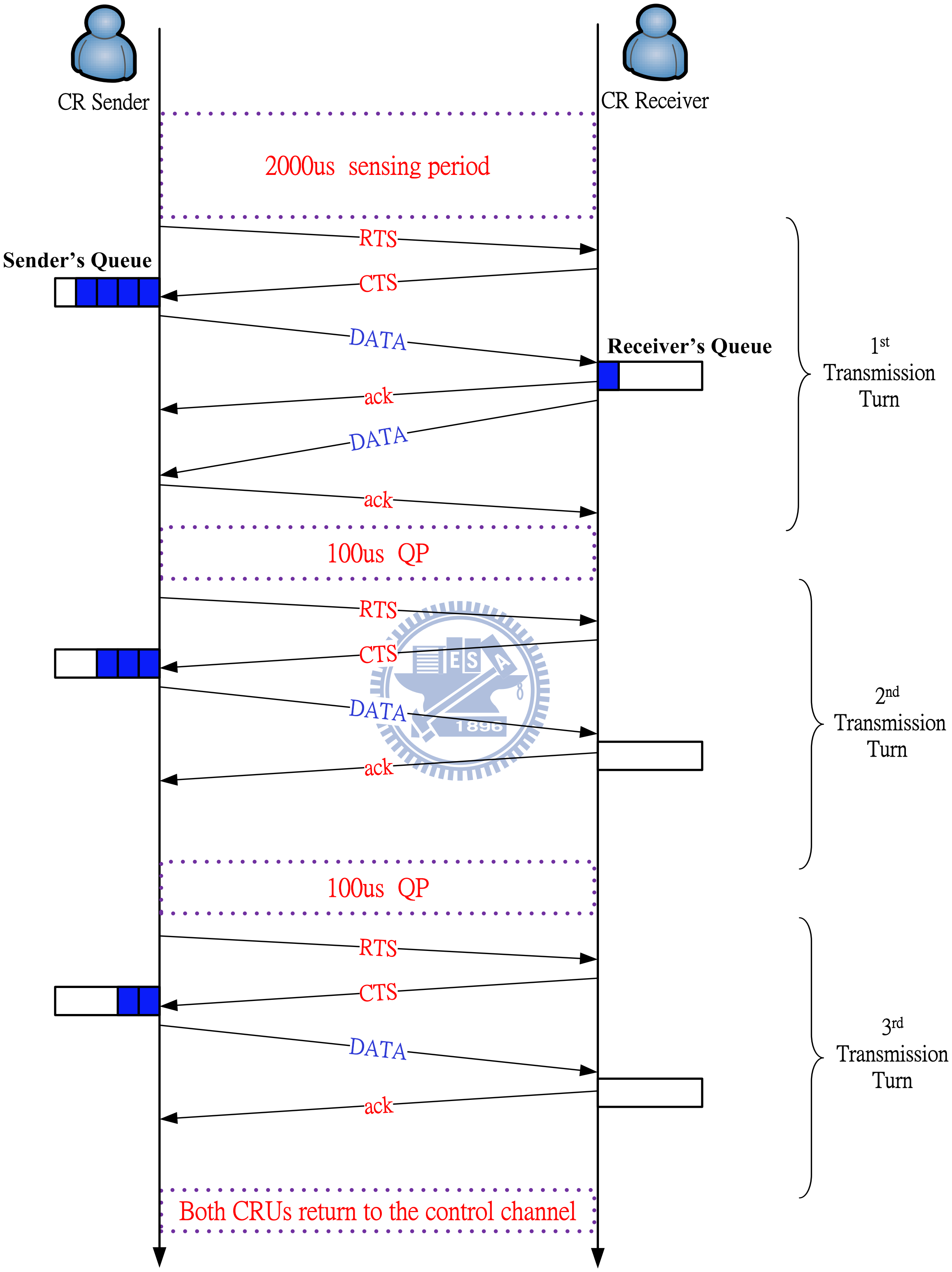


Figure 3.4: Illustration of Message Exchanges on Data Channel of BBi-MAC with Bi-directional Bandwidth Reservation when $Txop = 3$

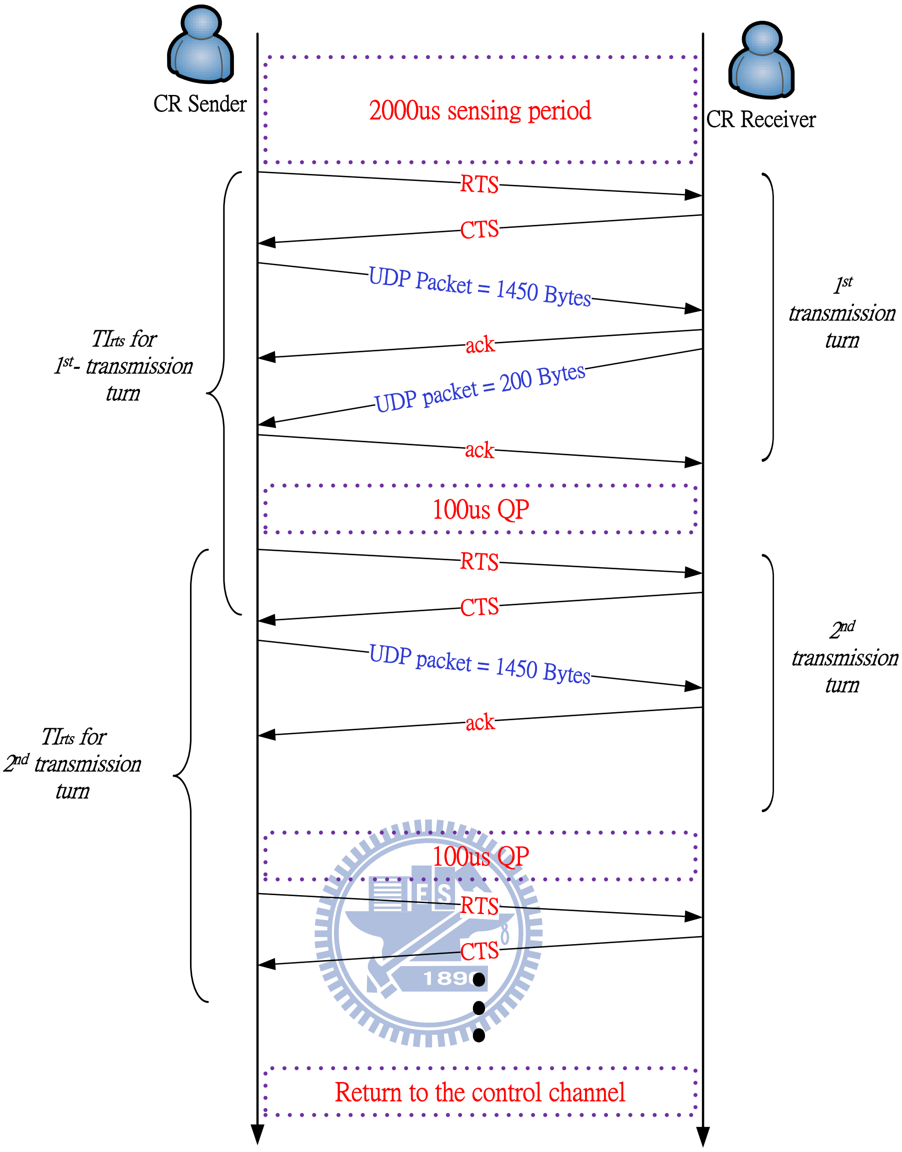


Figure 3.5: Illustration of inappropriate TI setting of BBi-MAC with Bi-directional Bandwidth Reservation

TI to be carried in the RTS control packet is determined solely based on the RT_{grt} value as elaborated in Section 3.2. However, when a CRU pair with asymmetry traffic flows are operating over the BBi-MAC, it is very likely that the situation as illustrated in Fig. 3.5 could happen.

Fig. 3.5 illustrates the negative impact which is caused by the TI setting in BBi-MAC when bi-directional bandwidth reservation is considered. In this example, the HOQ of the CR sender is an UDP packet with a size of 1450 Bytes. Also, the CR pair has reached a common agreement that a bi-directional bandwidth reservation is necessary because the CR receiver has a non-empty queue. With the current design of BBi-MAC, when transmitting an RTS control packet, the CR sender automatically sets its TI (TI_{rts}) to a duration which is adequate for transmitting two UDP packets of 1450 Bytes and two 802.11 ack packets.

From the figure, it is apparent that such an inappropriate TI setting can result in a negative impact when the traffic between the CR pair is asymmetry. This is because when others station, particularly the PUs, listen to the wireless medium and read the TI_{rts} as transmitted by the CR sender, they will set their NAV such that they know how long they must defer from accessing the medium. However, with asymmetry traffics, it is very likely that the CR receiver would send an UDP packet with smaller size (e.g., 200 Bytes) as illustrated in this figure. As a consequence, even when the CRU pair have finished their transmission turn and entered the QP, the PUs are still not able to claim the wireless channel because they are still in the process of deferring. From the continuous and overlapped portion of TI_{rts} for first and second transmission turn as shown in the figure, one can see that the channel is being monopolized by the CRU pair "un-intentionally". The same case happens when the CR receiver is running out of packets for the CR receiver, as shown in the CR receiver's second transmission turn 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 overheard TI_{rts}.

In either cases as elaborated above, the good intention of CRU pair entering the QP has been wasted because it does not provide the PUs any opportunity to claim the channel. Furthermore, one can see that eventually the CRU pair monopoly the data channel until they finish the *Txop-th* transmission turn. This is an undesirable situation which could severely affect the PUs' performances. This is because in the occurrence of such situations, the PUs could possibly grab the channel only after the CRU pair return to the control channel.

3.3.2 Design of Advance Scheme of Bi-directional CR MAC Protocol (ABi-MAC)

To address the two major deficiencies of BBi-MAC as unfolded in Section 3.3.1, in this section we propose an ABi-MAC, which serves as an extension of the BBi-MAC. The objective of the ABi-MAC is to better support the CRUs with asymmetry traffics, subjected that the PUs' performances are protected. We further divide the design of

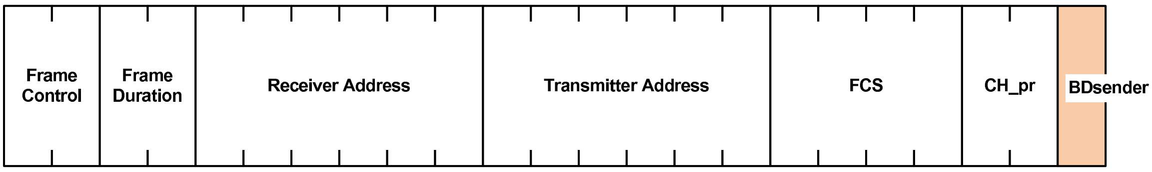


Figure 3.6: Frame Format of the control packet REQCR of ABi-CR

ABi-MAC into two phases as elaborated below.

Phase 1: Dynamic Bandwidth Assignment (DBA)

In phase one of the ABi-MAC, we propose a DBA such that the CRUs are not bounded to transmit only a pre-defined *Txop* packets. To this end, we define several new variations of *Txop*, namely MAX_PACKET, bandwidth demand of CR sender (BD_{sender}) and bandwidth demand of CR receiver ($BD_{receiver}$). Similarly, the MAX_PACKET is a pre-defined network parameter. However, instead of representing the number of packets that can be transmitted by each CRUs, it refers to the maximum number of packets that is allowable to be transmitted by a CRU pair when both CRUs are on the data channel. On the other hand, the BD_{sender} and $BD_{receiver}$ refer to the number of packets to be transmitted by the CR sender and CR receiver, respectively.

Considering that a CRN may carry asymmetry traffic flows between a CRU pair, the values of BD_{sender} and $BD_{receiver}$ are dynamically decided during the bandwidth negotiation procedure on the control channel. Therefore, the determination of BD_{sender} and BD_{sender} can reflect the instantaneous queue condition of the CRUs.

In the ABi-MAC, BD_{sender} and $BD_{receiver}$ are carried in the CR-related control packets, i.e., REQCR and GRANTCR, respectively. To this end, we replace the 2-bit field of *Reservation Type (RT)* in BBi-MAC with a 5-bit field of BD_{sender} or $BD_{receiver}$, as shown in Fig. 3.6 and Fig. 3.7, respectively. With the allocation of 5-bit field, each CRUs can request up to a maximum of 31 transmission opportunities. The remaining three bits are left unused for possible future extension.

The DTA of ABi-MAC operates in the following way. Firstly, we assume that a CR sender maintains a separate queue for every CR receiver. Before sending an REQCR, the CR sender has to scan through its queue to identify the number of packets to be transmitted for a specified CR receiver. This information is treated as the value of BD_{sender} in the REQCR. Same wise, upon receiving an REQCR, the CR receiver has to perform the

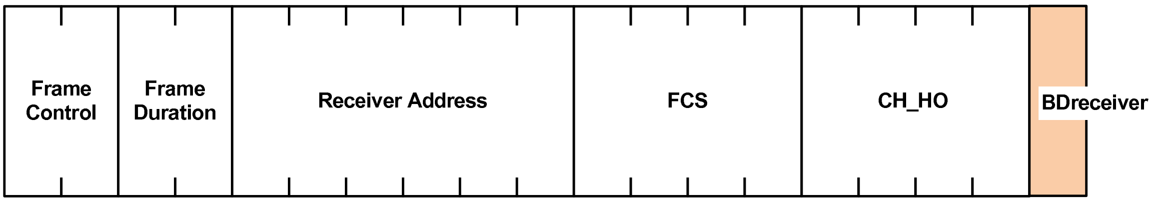


Figure 3.7: Frame Format of the control packet GRANTCR of ABi-CR

scanning procedure as described above and fill in the $BD_{receiver}$ value in the GRANTCR.

The principles of filling in the values of BD_{sender} and $BD_{receiver}$ are explained in Fig. 3.8, Fig. 3.9, Fig. 3.10 , Fig. 3.11 and Fig. 3.12. These figures show the possible values of BD_{sender} and $BD_{receiver}$ in different network traffic conditions. In all the demonstrated examples, we assume that *MAX_PACKET* is fixed to a value of 15. Also, we define the 'CR sender' as the CRU who initiates the bandwidth negotiation procedure by sending an REQCR, whereas the 'CR receiver' as the CRU who replies the CR sender with a GRANTCR.

In the first two examples as shown in Fig. 3.8 and Fig. 3.9, we consider network scenarios such that only one-way traffic flow is generated from the CR sender to the CR receiver. In ABi-MAC, when the CR sender triggers a bandwidth negotiation procedure, it sets the value of BD_{sender} using the following equation:

$$BD_s = \begin{cases} Q_s, & \text{if } Q_s \leq MAX \\ MAX, & \text{if } Q_s > MAX \end{cases} \tag{3.2}$$

where BD_s denotes the value of BD_{sender}, Q_s denotes the queue size of the CR sender and MAX denotes the pre-defined and fixed network parameter *MAX_PACKET*.

As shown in Fig. 3.8, since the number of packets in the queue of CR sender exceeds the pre-defined value of *MAX_PACKET* and a CRU is disallowed to transmit more than *MAX_PACKET* packets, the CR sender sets the value of BD_{sender} to *MAX_PACKET*, i.e., 15, according to Eq. 3.2. This value is determined based on the philosophy that a CR sender should always transmit as many packets as possible when it successfully 'steals' the data channel.

Upon receiving the REQCR, the CR receiver checks the corresponding queue for the CR sender before deciding the value of $BD_{receiver}$ using Eq. 3.3. At the same time, the CR receive also records and maintains the value of BD_{sender} as number of packets to be received from the sender (NPRs) using Eq. 3.4. As shown in Fig. 3.8, since the queue of

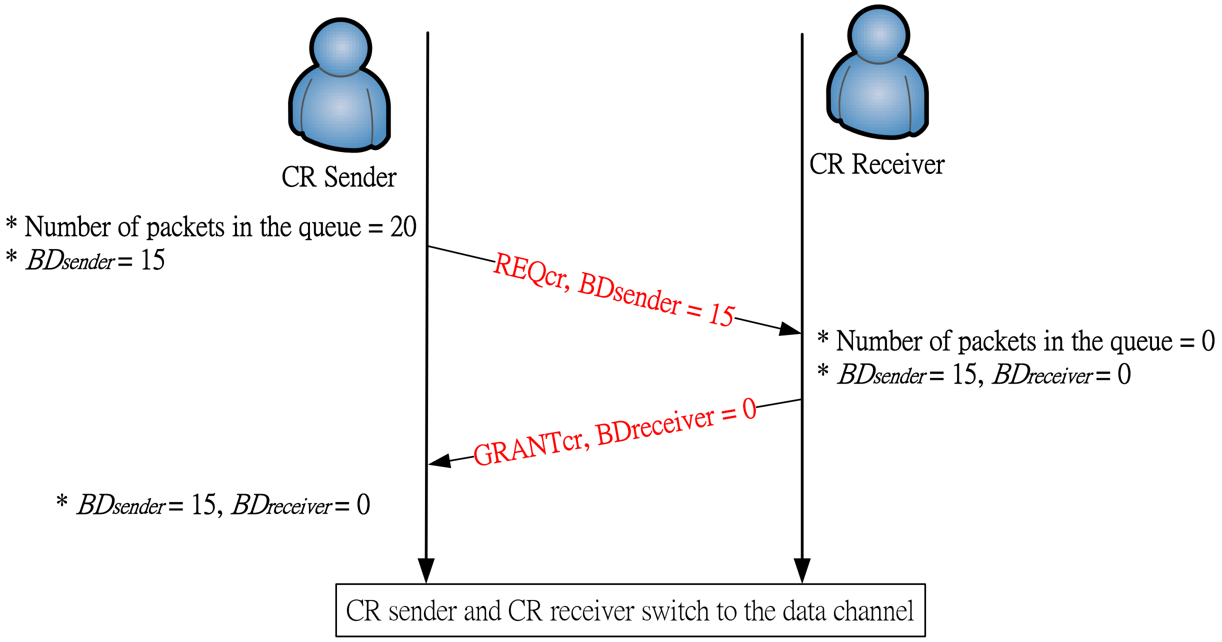


Figure 3.8: Possible values of BD_{sender} and $BD_{receiver}$ of ABi-MAC with uni-directional heavy traffic

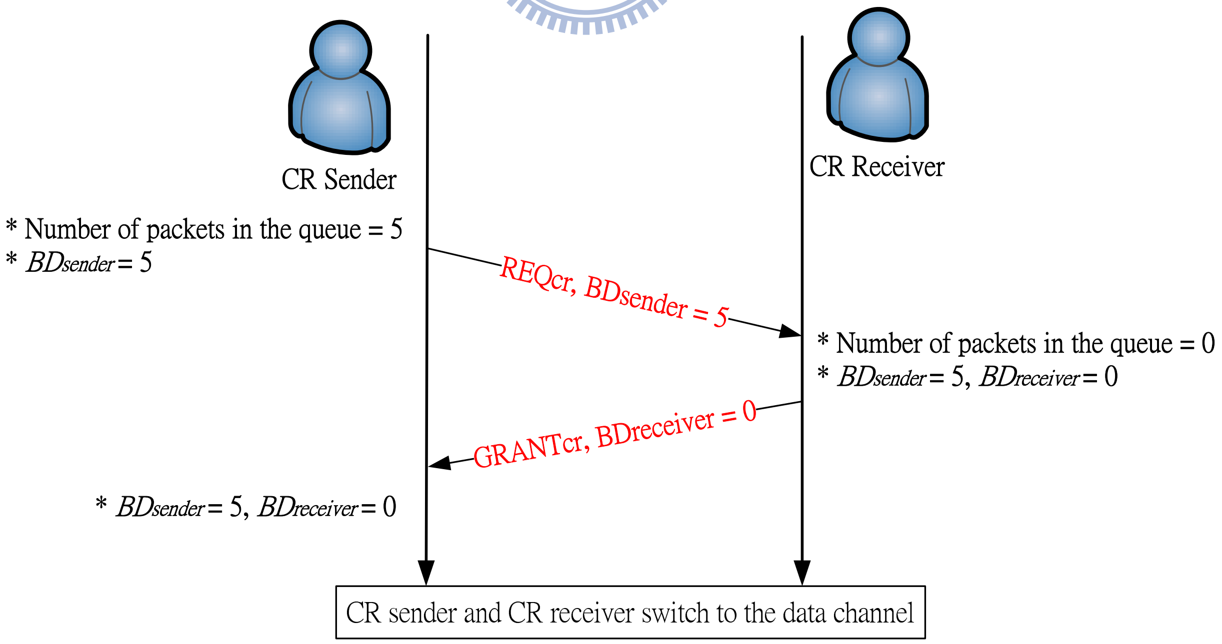


Figure 3.9: Possible values of BD_{sender} and $BD_{receiver}$ of ABi-MAC with uni-directional light traffic flow

the CR receiver is empty, the CR receiver sets the $BD_{receiver}$ to zero and records NPR$_{s}$ as 15. After that, it replies a GRANTCR to the CR sender.

$$BD_r = \begin{cases} 0, & \text{if } Q_r = 0 \\ MAX - BD_s, & \text{if } Q_r \geq \lfloor \frac{MAX}{2} \rfloor \text{ and } BD_s \leq \lceil \frac{MAX}{2} \rceil \\ \lfloor \frac{MAX}{2} \rfloor, & \text{if } Q_r \geq \lfloor \frac{MAX}{2} \rfloor \text{ and } BD_S \geq \lceil \frac{MAX}{2} \rceil \\ Q_r, & \text{if } Q_r \leq \lfloor \frac{MAX}{2} \rfloor \end{cases} \tag{3.3}$$

where BD_r denotes the value of BD_{sender} and Q_r denotes the queue size of CR receiver.

$$NPR_s = \begin{cases} BD_s, & \text{if } BD_r = 0 \\ MAX - BD_r, & \text{if } BD_r > 0 \end{cases} \tag{3.4}$$

Finally, the CR sender reads the value of $BD_{receiver}$ in GRANTCR and records the number of packets to be received from the receiver (NPR$_{r}$) using Eq. 3.5. It then finalizes its value of BD_{sender} using the Eq. 3.6. In this example, the CR sender confirms that the BD_{sender} remains to be 15. Therefore, it concludes that it has reached a common agreement with the CR receiver such that it can transmit up to 15 packets to the CR receiver when they are on the data channel.

$$NPR_r = \begin{cases} BD_s, & \text{if } BD_r = 0 \\ MAX - BD_r, & \text{if } BD_r > 0 \end{cases} \tag{3.5}$$

$$BD_{SF} = \begin{cases} BD_S, & \text{if } BD_R == 0 \\ MAX - BD_R, & \text{if } BD_R \leq \lfloor \frac{MAX}{2} \rfloor \text{ and } BD_S \geq \lceil \frac{MAX}{2} \rceil \\ BD_S, & \text{if } BD_R \geq \lfloor \frac{MAX}{2} \rfloor \text{ and } BD_S \leq \lceil \frac{MAX}{2} \rceil \end{cases} \tag{3.6}$$

where BD_{SF} denotes the finalized value of BD_{sender} at the end of the bandwidth negotiation procedure.

It is worth to note that in our design of DBA, a CRU pair are required to keep track of the number of transmitted packets and the number of received packets whenever they are on the data channel. By doing so, they can always correctly determine the right timing of swapping their identities or returning to the control channel which will be explained later.

In example two, the CR sender in Fig. 3.9 only generates very light traffic. At the end of the bandwidth negotiation procedure, the CR sender will be allocated uni-directional bandwidth to transmit five packets, which exactly meets its needs of the bandwidth

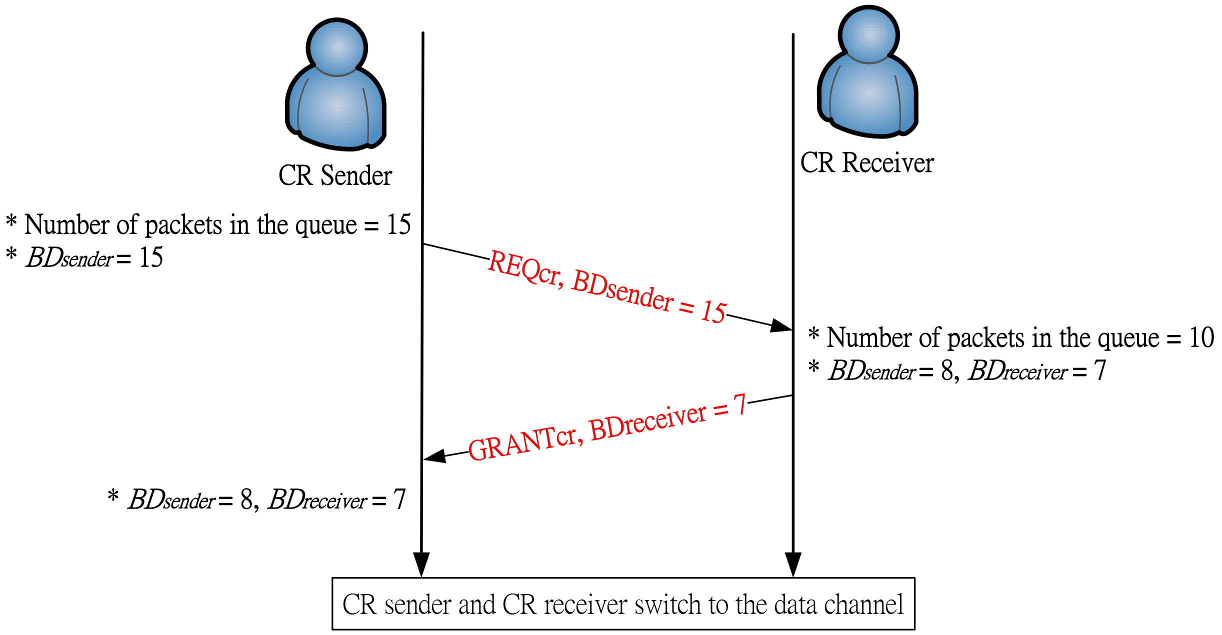


Figure 3.10: Possible values of BD_{sender} and $BD_{receiver}$ of ABi-MAC when the traffic between a CRU pair is heavy

usage. This shows that the DBA of ABi-MAC offers very high flexibility in terms of the bandwidth allocation since the CRUs are not longer bounded to transmit $Txop$ packets as the case of BBi-MAC. Instead, the bandwidth can be adaptively allocated to the CRUs depends on their instantaneous bandwidth demand.

In example three, we consider a network scenario such that both the CR sender and CR receiver are generating heavy traffic for one another as shown in Fig. 3.10. Following the Eq. 3.2, the CR sender sets the value of BD_{sender} to MAX_PACKET, i.e., 15, upon transmitting an REQCR. On the receiver side, when the CR receiver checks the corresponding queue for the CR sender, it finds that the number of packets being queued exceeds $\lfloor \frac{MAX\_PACKET}{2} \rfloor$. Therefore, the value of $BD_{receiver}$ is set to $\lfloor \frac{MAX\_PACKET}{2} \rfloor$, i.e., seven, according to Eq. 3.3. Such a setting is determined based on the philosophy that the CRU pair should access the bandwidth in a fair manner when both have high demand in using the bandwidth.

In example four, we still consider a network scenario with a CRU pair generating traffic for one another as shown in Fig. 3.11. Different from example three, now we assume that the CR receiver is generating a heavier traffic to the CR sender as compared to the opposite traffic from the CR sender. Before sending an REQCR, the value of BD_{sender} is set to the queue size of the CR sender, i.e., five, according to Eq. 3.2.

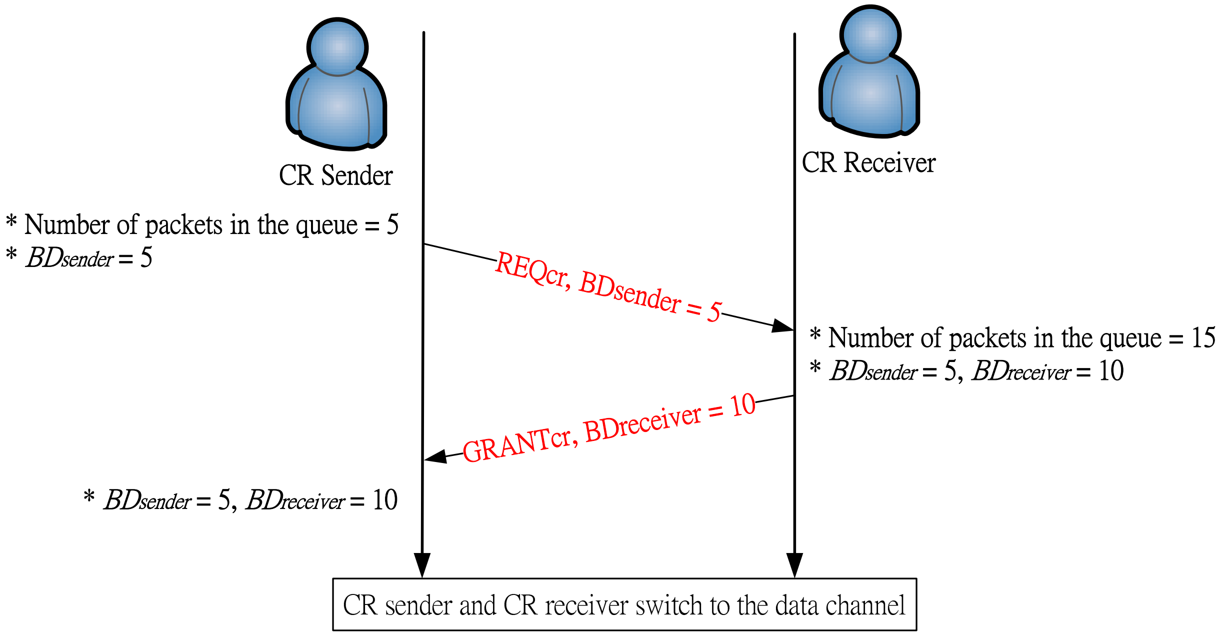


Figure 3.11: Possible values of BD_{sender} and $BD_{receiver}$ of ABi-MAC when the traffic of CR receiver is heavier than that of CR sender

Since the *MAX_PACKET* is pre-defined to 15, ideally the CRU pair should take the good opportunity to transmit up to 15 packets if possible when they are on the data channel. Therefore, following Eq. 3.3, the CR receiver can smartly use the left over transmission chances and set the $BD_{receiver}$ to a value of ten. In this case, bi-directional packets transmission will take place until the fifth transmission turn. Starting from the sixth transmission turn until the tenth transmission turn, the protocol degenerates to uni-directional packet transmission, from the 'CR receiver' to the 'CR sender'. As compared to BBi-MAC, this design can maximize the throughput of CRUs as explained below.

In BBi-MAC, a CR sender is assumed to transmit exactly *Txop* packets when it is on the data channel. However, when the queue size of a CR sender (denoted as CRU A) is less than *Txop*, there is no mechanism for CRU A to inform the CR receiver (denoted as CRU B). As a result, when the queue of CRU A becomes empty, it switches back to the control channel without notifying CRU B. On the other hand, after the QP, CRU B continues to wait for the packets from CRU A. When the timer expires, it assumes that CRU A has detected the return of PUs and hence switched to the next data channel as pre-negotiated. CRU B thus also switches to that data channel and repeats the same actions as described earlier. This shall be carried on until CRU B has visited all the data channels but still not receiving any packets from CRU A. Eventually, it will return to the control channel.

If both the CRUs are running out of packets for delivery, the time spent on waiting for the packets from the CRU A will not result in any negative impact. However, if the CRU A has new packets for the CRU B during that duration, it finds that it cannot complete any bandwidth negotiation procedure with CRU B as CRU B is away from the control channel. Similarly, if CRU B has queued packets for CRU A, these packets cannot be transmitted until CRU B returns to the control channel and successfully initiates a bandwidth negotiation with CRU A. Obviously, all the above mentioned negative impacts can affect the achievable throughputs of CRUs in a significant way.

In ABi-MAC, the above problems could be effectively avoided through the proposed DBA mechanism. This is because the CRU pair are allowed to switch their identities (i.e., sender and receiver) according to their traffic load even when they are on the data channel. For instance, in the example four, initially CRU A plays the role as a CR sender, thus it initiates RTS/CTS handshake before every packet transmission. After the fourth transmission turn, however, the CRU A swaps its identity with CRU B because it has no more packets to be transmitted to CRU B.

With the maintained information, a CRU pair can always determine the right timing of returning to the control channel or swapping their identities. Generally, a CRU pair return to the control channel synchronously when the following two conditions are met:

- For CR sender: The number of received packets equals to $\mathrm{NPR}_{\mathrm{r}}$ and the number of transmitted packets equals to BD_{sender}.

- For CR receiver: The number of received packets equals to $\mathrm{NPR}_{\mathrm{s}}$ and the number of transmitted packets equals to $BD_{receiver}$.

On the other hand, a CRU pair swap their identities when the following two conditions are met:

- For CR sender: The number of transmitted packets equals to BD_{sender} but the number of received packets less than $\mathrm{NPR}_{\mathrm{r}}$.

- For CR receiver: The number of transmitted packets less than $BD_{receiver}$ but the number of received packets equals to $\mathrm{NPR}_{\mathrm{s}}$.

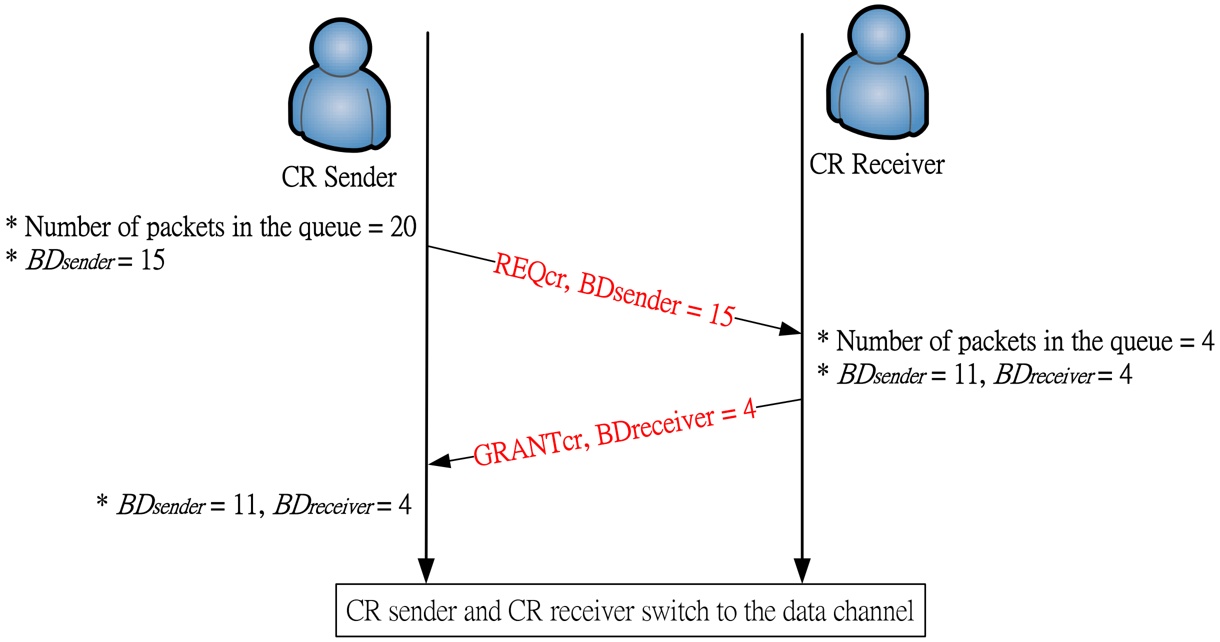


Figure 3.12: Possible values of BD_{sender} and $BD_{receiver}$ of ABi-MAC when the traffic of CR sender is heavier than that of CR receiver

In the example above, after the forth transmission turn, both the CRUs meets the conditions of swapping their identities. Therefore, starting from the fifth transmission turn, CRU B will be the CR sender and thus initiates the RTS/CTS handshake with CRU A. Through such design, a CRU pair can always maximize the achievable throughput once they gain access to the data channel. This is because a CRU pair can always minimize the overhead caused by switching from the data channel to control channel and vice versa, especially when the traffic load between a pair of CRU is asymmetry.

Finally, Fig. 3.12 demonstrates a similar network scenario as example four, except that the CR sender is generating a heavier traffic to the CR receiver. Again, one can see that through our proposed DBA mechanism, the sum of packets to be transmitted by the CRU pair is being maximized to MAX_PACKET. In this example, the CR sender remains as the RTS initiator throughout the duration when it is on the data channel. Bi-directional packets transmission will first take place until the fourth transmission turn. Starting from the fifth transmission turn until the 11-th transmission turn, again the protocol degenerates to uni-directional packets transmission.

To sum up, from all the demonstrated examples, we show that the proposed DBA of ABi-MAC can offer very high flexibility in allocating bandwidth to the CRU pair adaptively. Regardless of the traffic conditions between a CRU pair, the DBA can well support every CRU based on its spontaneous traffic load. As a result, the ABi-MAC can maximize the achievable throughput in a CRN by minimizing the overhead caused by switching from the control channel to data channel and vice versa.

Phase 2: Smart Transaction Interval (TI) Setting

In phase two of the ABi-MAC, we propose a Smart Transaction Interval Setting (STIS) such that the transaction information (TI) to be carried in the control packets can provide the neighboring PUs correct timing information. This is to counteract the negative impact which is caused by the inaccurate TI setting in BBi-MAC when bi-directional asychronous traffic flows are considered.

In a typical 802.11 network, a two-way handshake of RTS/CTS is adopted for bandwidth reservation as shown in Fig. 3.13. In the process of handshaking, the TI is carried along the RTS packet to indicate how long a sender wants to hold the medium. The TI of the RTS packet (TI_{rts}) is set based on the following equation:

$$TI_{rts} = D_{cts} + D_{data} + D_{ack} + 3 * sifs. \tag{3.7}$$

where D_{cts}, D_{data} and D_{ack} denotes the duration to transmit a CTS, data and 802.11 packet, respectively, and $sifs$ denotes the sifs interval.

In return, the receiver replies with a CTS packet echoing the expected duration of transmission. The TI of a CTS packet (TI_{cts}) is set based on the following equation:

$$TI_{cts} = TI_{rts} - sifs - D_{cts}. \tag{3.8}$$

Through the exchange of RTS and CTS control packets, all the nodes within hearing distance of either the sender or receiver or both will set their NAV according to the TI of overheard packets. Generally, NAV refers to the counter to be maintained by every station and it is the amount of time that must elapse until the medium will become free again. A station should only start to sense the channel for data transmission when its NAV is decremented to zero.

In BBi-MAC, the problem arose in the presence of bi-directional traffic because the RTS/CTS two-way handshake is inadequate to correctly provide TI information to the neighboring PUs of a CRU pair. As a result, the right of the neighboring PUs in using the data channel might be seriously violated because they will miss most of the QPs.

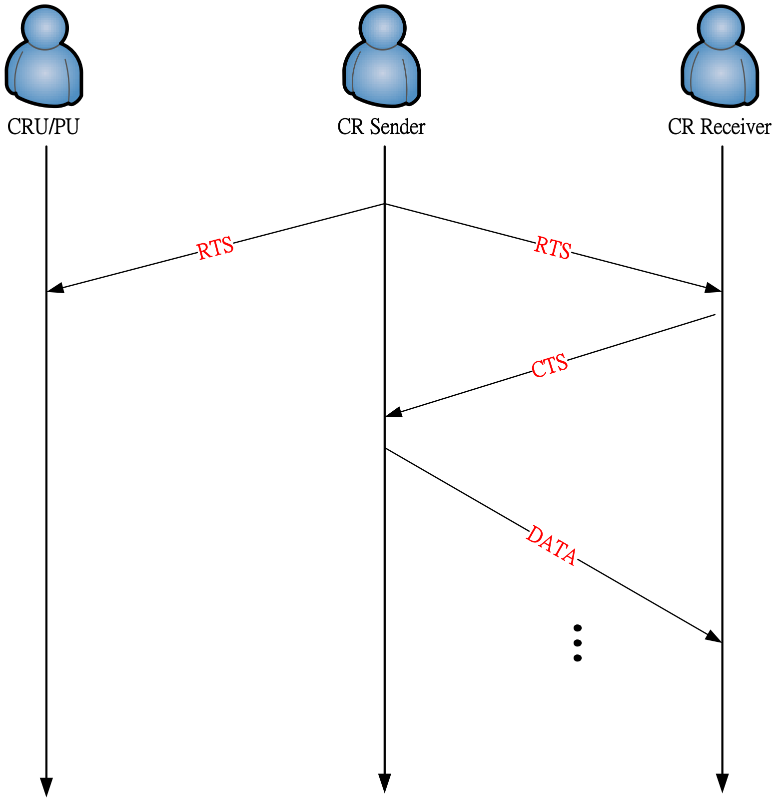


Figure 3.13: Illustration of the conventional RTS-CTS two-way handshake

Therefore, in the design of STIS, we introduce a conditional RTS/CTS/RTS$_e$ three-way handshake. This three-way handshake has to be performed by the CRU pair only when bi-directional bandwidth reservation is needed.

In the presence of bi-directional packets transmission, a CR receiver shall include the expected time to transmit the reverse data packet in the TI information that is carried along the CTS packet. As such, those neighboring PUs who are within the hearing distance of the CR receiver will defer their transmission until the TI is over. However, PUs who are located near to the CR sender but outside the hearing distance of the CR receiver will not be updated with the TI. Consequently, the right of these neighboring PUs in using the data channel might be seriously violated. Furthermore, the achievable throughput of the CRUs might be degraded due to potential packets collision.

We take the network topology as illustrated in Fig. 3.14 as an example. In this example, a CRU pair and several PUs coexist on a data channel and bi-directional packets transmission are taking place between the CRU pair. We assume that the PU A and PU B are within the transmission range of CRU A but outside the hearing range of CRU B. Before a data transmission, the CRU pair perform the RTS/CTS two-way handshake as

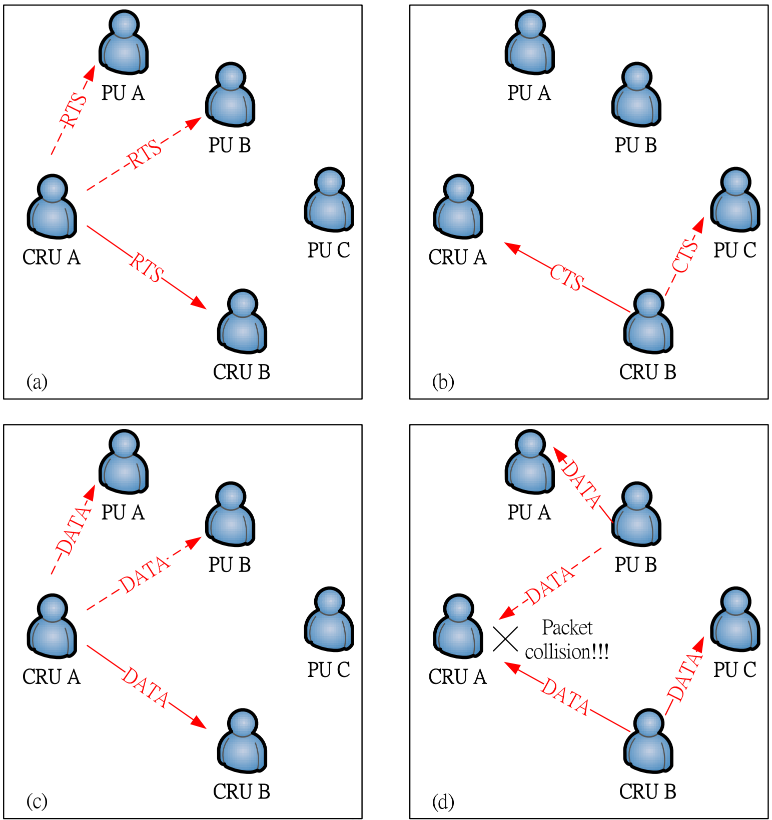


Figure 3.14: Illustration of packet collisions at the CRU B due to incorrect TI_{rts}

shown in Fig. 3.14(a) and (b). However, since the CR sender has no information about the length of the reverse packet, it is impossible for it to include the reverse packet's transmission time in the TI_{rts}. Therefore, the TI_{rts} is calculated simply based on Eq. 3.7.

Therefore, at the end of the two-way handshake, only PU C is being updated with the latest TI through the overheard CTS packet. When the data transmission from CRU A to CRU B is taking place as shown in Fig. 3.14(c), all the PUs will keep silent due to their NAV which was set based on the TI_{rts} or TI_{cts}. However, a problem arises when the TI_{rts} is expired and PU A starts a data transmission to PU B while CRU B is sending a reverse packet as shown in Fig. 3.14(c). This could happen because PU A cannot hear the data transmission of CRU B and thus senses the channel as idle. As a result, the packets transmitted by PU A will collide with those from CRU B at CRU A, thus degrading the achievable goodputs of CRUs.

Fig. 3.15 demonstrates another bad impact of incorrect TI_{rts} information. In this example, the problem arises when PU A intends to start a data transmission after a QP but senses that the channel is busy due to the transmission of CRU A. It is known that the PUs in 802.11 network operates based on the CSMA/CA MAC protocol. With this

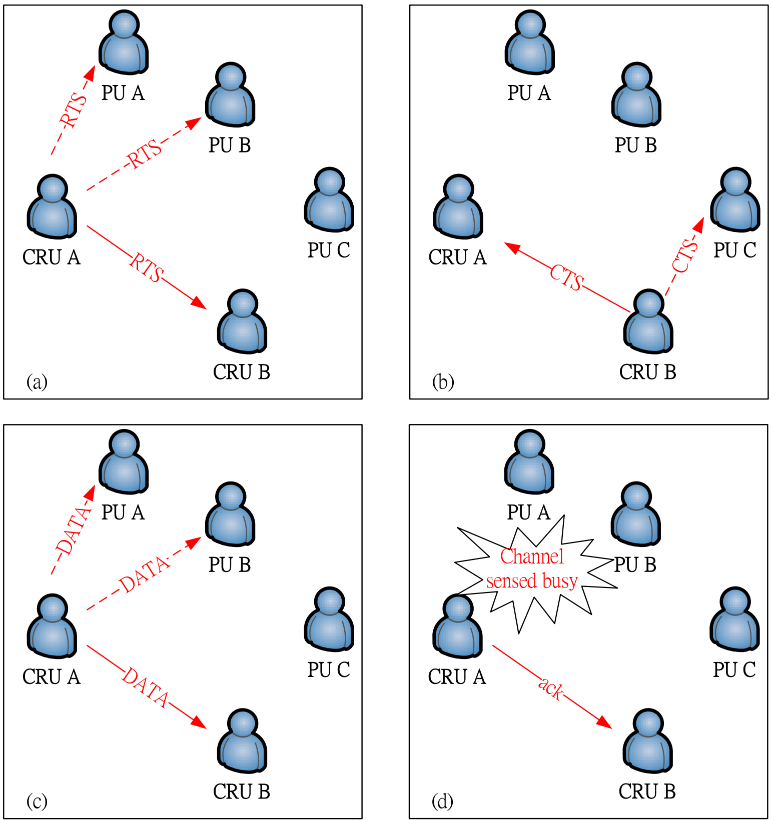


Figure 3.15: Illustration of the PUs' right being violated due to incorrect TI_{rts}

protocol, a PU shall sense the medium whenever it wishes to transmit a packet. If the medium is sensed idle, it defers a DIFS gap and then transmits the packet. In contrast, if the medium is sensed busy, it defers a DIFS gap and then regenerates random backoff period within the contention window for an additional deferral time as a mean to resolve contention. Therefore, any occurrence of the incident as shown in Fig. 3.15(d) will cause the PUs to defer for unnecessary extra time before it can re-access the channel for data transmission. Therefore, the PUs' right of using the data channel is violated.

To solve the above two problems, we propose a conditional transmission of RTS_e by the CR sender after receiving a CTS. As such, the CR sender can effectively update all of its neighboring PUs with its latest TI whenever bi-directional bandwidth reservation is required. The newly introduced RTS_e control packet in ABi-MAC is identical to the conventional RTS packet, except that its TI has to be set based on the following equation:

$$TI_{rts_e} = TI_{cts} - sifs - D_{cts}. \tag{3.9}$$

In the RTS/CTS/RTS_e three-way handshake, the TI of a CTS packet is set according to the bandwidth reservation type using the following equation:

$$TI_{cts} = \begin{cases} TI_{rts} - sifs - D_{cts}, & \text{if uni-directional bandwidth reservation} \\ TI_{rts} - D_{cts} + D_{rts}, & \text{if bi-directional bandwidth reservation} \end{cases} \tag{3.10}$$

Upon receiving a CTS packet, if a bi-directional bandwidth reservation is required for the upcoming transmission turn, the CR sender immediately transmits an RTS_e control packet with the TI_{rtse} calculated based on Eq. 3.9. Fig. 3.16 illustrates the packets exchange of a CRU pair when RTS/CTS/RTS_e three-way handshake takes place in the presence of bi-directional traffic flows. In general, the upcoming bandwidth reservation type at a CRU can be determined easily based on the maintained records as follows:

- For CR sender (before sending an RTS_e control packet)
  - Bi-directional bandwidth reservation: The number of transmitted packets less than BD_{sender} and the number of received packets less than NPR_r.
  - Uni-directional bandwidth reservation: The number of transmitted packets less than BD_{sender} and the number of received packets equals to NPR_r.
- For CR receiver (before sending a CTS control packet)
  - Bi-directional bandwidth reservation: The number of transmitted packets less than $BD_{receiver}$ and the number of received packets less than NPR_s.
  - Uni-directional bandwidth reservation: The number of transmitted packets equals to than $BD_{receiver}$ and the number of received packets less than NPR_r.

To sum up, through the proposed STIS, ABi-MAC can smartly broadcast the correct information of TI to all the neighboring nodes of a CRU pair. A CRU pair performs the conventional RTS/CTS two-way handshake when the packets to be transmitted are uni-directional. In contrast, the CRU pair performs the proposed RTS/CTS/RTS_e three-way handshake in the presence of bi-directional traffic. By correctly setting the TI values through the proposed STIS, ABi-MAC eliminates the occurrence of PUs sensing a busy channel which is caused by the CRUs and thus provides a better protection to PUs. Also, the goodputs of CRUs can be improved through reducing potential packets collision.

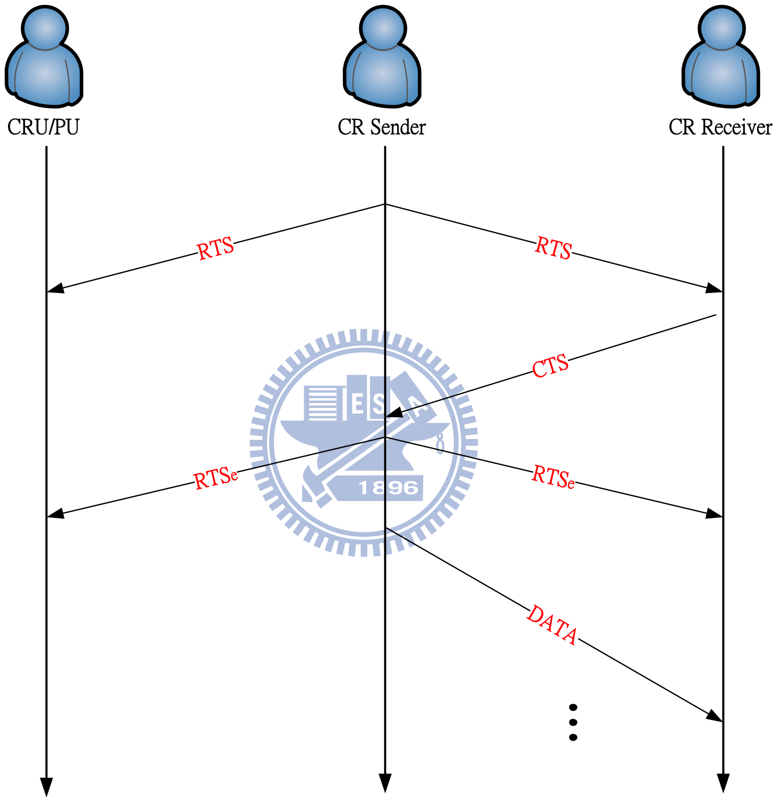


Figure 3.16: Illustration of RTS-CTS-RTS$_e$ three-way handshake

Chapter 4

Performance Evaluation and Discussion

In this chapter, we used both the mathematical analysis and simulations to evaluate the performances of BBi-MAC and ABi-MAC. First of all, mathematical analysis of Uni-MAC and BBi-MAC in a simple network scenario with only a CRU pair is presented. The performance gain of BBi-MAC as shown in the mathematical analysis greatly convinced the idea of our proposed BBi-MAC. Follow on, we used simulation results to verify our mathematical analysis. Lastly, through extensive simulations, we showed the performance enhancement of our proposed BBi-MAC and ABi-MAC as compared to the benchmark protocol, Uni-MAC, in a more complex network scenarios.

4.1 Mathematical Analysis of the Achievable Throughputs of Uni-MAC and BBi-MAC

In this section, the average throughput of a CRU pair when the CRUs are operating using either Uni-MAC or BBi-MAC is analyzed. The analysis presented also demonstrates the impact of *Txop* setting on the investigated protocols' performances. This mathematical analysis greatly helps us to gain more insight into the performance difference between the Uni-MAC and BBi-MAC, and thus convinced our idea of the proposed protocol modifications on Uni-MAC.

In this analysis, only a simple network with a pair of CRUs is investigated. The considered traffic types include: 1) single TCP flow, and 2) bi-directional UDP flows.

Table 4.1: Fixed Network Parameters

Parameter Name	**Symbol**	**Value**
Channel Bandwidth	Bw	2 Mbps
Time to transmit a preamble	$T_{preamble}$	144 us
Time to transmit PLCP header	T_{plcp}	48 us
Duration of SIFS	SIFS	10 us
Duration DIFS	DIFS	10 us
Duration of QP	QP	100 us
Packet Length of TCP Payload	LEN_{tcp}	1448 Bytes
Packet Length of TCP ACK	LEN_{ack}	68 Bytes
Packet Length of UDP Payload	LEN_{udp}	1450 Bytes
Packet Length of Control Packets	LEN_{ctrl}	20 Bytes
Length of MAC Header	LEN_{hr_mac}	28 Bytes
Length of IP Header	LEN_{hr_ip}	20 Bytes
Length of TCP Header	LEN_{hr_tcp}	20 Bytes
Length of UDP Header	LEN_{hr_udp}	20 Bytes

The objective of this analysis is to find out the maximum achievable throughput of a CRU pair in transmitting either a TCP flow or two bi-directional UDP flows. To simplify the analysis, it is assumed that all the control packets (i.e., REQCR, GRANTCR, RTS, CTS, 802.11 ack) have an equal packet length and all the CRUs defers for an equal RWD of 50 microseconds before transmitting an REQCR. In Table 4.1, those fixed parameters to be used in this analysis are listed.

To calculate the achievable throughput, transmission times of various packet types have to be known in advance. Eq. 4.1 shows the transmission time of a packet across a CRN, where LEN denotes the packet's length in unit of Byte. Using this equation, a packet's transmission time can be calculated simply by replacing the LEN with that packet's length as listed in Table 4.1. Nevertheless, when TCP or UDP payload is being transmitted over a network, various headers are appended to the payload. Therefore, all these necessary headers (i.e., TCP/UDP header, IP header and MAC header) should be taken into consideration when the transmission time of a TCP data or an UDP data is calculated. Table 4.2 displays the calculated transmission time of various packet types

Table 4.2: Transmission time of different types of packet

Packet Type	**Symbol**	**Total length (Bytes)**	**Transmission Time (us)**
Control Packet	T_{CTRL}	14	248
TCP ACK	T_{ACK}	68	464
TCP DATA	T_{TCP}	1516	6256
UDP DATA	T_{UDP}	1518	6264

using Eq. 4.1.

$$\mathrm{T} = \frac{LEN * 8}{Bw * 10^6} * 10^6 + T_{preamble} + T_{plcp}. \tag{4.1}$$

Furthermore, to simplify our mathematical model, we regard an "effective TCP data" as the the combination transmissions of a TCP data and a TCP ACK. This is because for TCP flows, the replied TCP ACKs deos not add value to the TCP throughput. Therefore, TCP ACKs should not be treated as a TCP data. However, for UDP flows, since the ACK policy is not being employed for congestion control, every successful UDP packet transmission is regarded as an effective UDP data.

First of all, taking $Txop = 1$ and knowing the transmission time of different packet types, the time to be spent by a CRU pair for transmitting an effective TCP data using Uni-MAC is divided and categorized as below:

1. Bandwidth negotiation procedures on the control channel by TCP sender, T_{bnp}:
    - TCP sender waits for a RWD.
    - TCP sender transmits an REQCR
    - TCP receiver performs fast sense on all the five data channels
    - TCP receiver transmits a GRANTCR
2. Channel sense on the data channel, T_{sense}
    - The CRU pair sense the channel for 2000 us
3. RTS/CTS handshake on the data channel, T_{hs}:
    - TCP sender defers for sifs duration

    - TCP sender transmits a RTS control packet
    - TCP receiver defers for sifs duration
    - TCP receiver transmits a CTS control packet

4. Data transmission on the data channel by TCP sender, T_s:
    - TCP sender defers for difs duration
    - TCP sender transmits a TCP data packet
    - TCP receiver defers for a sifs duration
    - TCP receiver transmits an 802.11 ack
5. Bandwidth negotiation procedures on the control channel by TCP receiver, T_{bnp}:
    - TCP receiver waits for a RWD
    - TCP receiver transmits an REQCR
    - TCP sender performs fast sense on all the five data channels
    - TCP sender transmits a GRANTCR
6. Channel sense on the data channel, T_{sense}:
    - The CRU pair sense the channel for 2000 us
7. RTS/CTS handshake on the data channel, T_{hs}:
    - TCP receiver defers for sifs duration
    - TCP receiver transmits a RTS control packet
    - TCP sender defers for sifs duration
    - TCP sender transmits a CTS control packet
8. Data transmission on the data channel by TCP receiver, T_r:
    - TCP receiver defers for difs duration
    - TCP receiver transmits a TCP ACK packet
    - TCP sender defers for a sifs duration
    - TCP sender transmits an 802.11 ack

From the items as listed above, T_{bnp}, T_{hs}, T_s and T_r can be calculated using the equations as shown below:

$$T_{bnp} = RWD + T_{ctrl} + 500 + T_{ctrl}. \tag{4.2}$$

$$T_{hs} = sifs + T_{ctrl} + sifs + T_{ctrl}. \tag{4.3}$$

$$T_s = \begin{cases} disf + T_{TCP} + sifs + T_{ctrl}, & \text{if TCP DATA} \\ disf + T_{UDP} + sifs + T_{ctrl}, & \text{if UDP DATA} \end{cases} \tag{4.4}$$

$$T_r = disf + T_{ACK} + sifs + T_{ctrl}. \tag{4.5}$$

When considering *Txop* with a value of larger than one, the time duration that a CRU pair enter a QP after every successful packet transmission should be included. Eq. 4.6 shows the effective time to transmit *Txop* effective TCP data under different *Txop* settings. Using Eq. 4.6, the theoretically achievable TCP throughput of a CRU pair under different *Txop* can be calculated using Eq. 4.7.

$$T_{tcp\_eff\_uni} = \begin{cases} 2(T_{bnp} + T_{sense} + T_{hs}) + T_s + T_r, & \text{if } Txop = 1 \\ 2(T_{bnp} + T_{sense}) + Txop(2 * T_{hs} + T_s + T_r + QP), & \text{if } Txop > 1 \end{cases} \tag{4.6}$$

$$THP_{tcp\_uni} = \frac{LEN_{tcp} * 8 * Txop}{T_{tcp\_eff\_uni} * 10^{-6}}. \tag{4.7}$$

Unlike TCP flows, in the calculation of the transmission time of an effective UDP data, the T_r can be omitted. Eq. 4.8 and Eq. 4.9 shows the required time to transmit effective *Txop* UDP data and the theoretically achievable UDP throughput of a CRU pair, respectively.

$$T_{udp\_eff\_uni} = \begin{cases} T_{bnp} + T_{sense} + T_{hs} + T_s, & \text{if } Txop = 1 \\ T_{bnp} + T_{sense} + Txop(T_{hs} + T_s + QP), & \text{if } Txop > 1 \end{cases} \tag{4.8}$$

$$THP_{udp\_uni} = \frac{LEN_{udp} * 8 * Txop}{T_{udp\_eff\_uni} * 10^{-6}}. \tag{4.9}$$

Next, the achievable throughput of a CRU pair when they are operating using BBi-MAC is analyzed. Again, the time to be spent by a CRU pair for transmitting an effective TCP data (or two effective UDP data) is divided and categorized as below:

1. Bandwidth negotiation procedures on the control channel by CR sender, T_{bnp}:
    - CR sender waits for a RWD.
    - CR sender transmits an REQ$_{\mathrm{CR}}$
    - CR receiver performs fast sense on all the five data channels
    - CR receiver transmits a GRANT$_{\mathrm{CR}}$
2. Channel sense on the data channel, T_{sense}
    - The CRU pair sense the channel for 2000 us
3. RTS/CTS handshake on the data channel, T_{hs}:
    - CR sender defers for sifs duration
    - CR sender transmits a RTS control packet
    - CR receiver defers for sifs duration
    - CR receiver transmits a CTS control packet
4. Data transmission on the data channel, $T_{transmit}$:
    - CR sender defers for difs duration
    - CR sender transmits a TCP DATA (UDP DATA)
    - CR receiver defers for a sifs duration
    - CR receiver transmits an 802.11 ack
    - CR receiver defers for a sifs duration
    - CR receiver transmits a TCP ACK (UDP DATA)
    - CR sender transmits an 802.11 ack

Different from the Uni-MAC, one can see that in BBi-MAC, the transmission time of an effective TCP data has been greatly reduced because the TCP receiver can immediately replies a TCP ACK using the reserved bandwidth in the same transmission opportunity. Therefore, the new parameter $T_{transmit}$ is calculated using Eq. 4.10. Eq. 4.11 shows the required transmission time to deliver *Txop* effective TCP data or *2*Txop* effective UDP data under different *Txop* settings. Lastly, the theoretically achievable TCP throughput and UDP throughput can be calculated using Eq. 4.12 and Eq. 4.13, respectively.

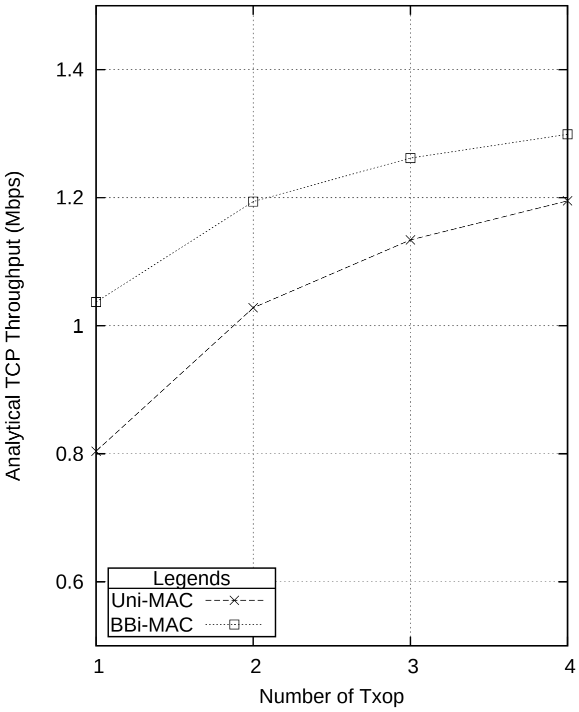


Figure 4.1: Analytical TCP Throughputs of Uni-MAC and BBi-MAC

$$T_{transmit} = \begin{cases} disf + T_{TCP} + sifs + T_{ctrl} + sifs + T_{ACK} + sifs + T_{ctrl}, & \text{if TCP DATA} \\ disf + T_{UDP} + sifs + T_{ctrl} + sifs + T_{UDP} + sifs + T_{ctrl}, & \text{if UDP DATA} \end{cases} \tag{4.10}$$

$$T_{tcp\_eff\_bbi} = \begin{cases} T_{bnp} + T_{sense} + T_{hs} + T_{transmit}, & \text{if } Txop = 1 \\ T_{bnp} + T_{sense} + Txop(T_{hs} + T_{transmit} + QP), & \text{if } Txop > 1 \end{cases} \tag{4.11}$$

$$THP_{tcp\_bbi} = \frac{LEN_{tcp} * 8 * Txop}{T_{tcp\_eff\_bbi} * 10^{-6}}. \tag{4.12}$$

$$THP_{udp\_bbi} = \frac{LEN_{tcp} * 8 * Txop * 2}{T_{udp\_eff\_bbi} * 10^{-6}}. \tag{4.13}$$

In Fig. 4.1 and Fig. 4.2, the analytical TCP throughput and UDP throughput for a CRU pair are presented. As depicted in these figures, one can see that the BBi-MAC outperforms the Uni-MAC in both the TCP throughput and UDP throughput even when the considered CRN consists of only a pair of CRU. These analytical results will be verified by our simulation results which will be presented in Section 4.3.

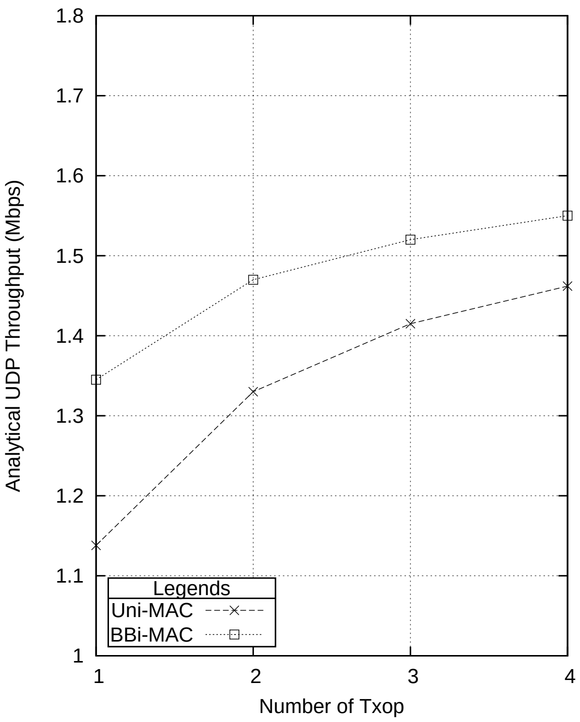


Figure 4.2: Analytical UDP Throughputs of Uni-MAC and BBi-MAC

4.2 Simulation Setup

In this section, we introduce the simulation setup for evaluating the proposed BBi-MAC and ABi-MAC. The NCTUns network simulator and emulator [16] is used for the evaluation. Table 4.3 shows the parameter setting used in our simulations. A * sign next to a value indicates that the value is used by default when the parameter is not varied as a control variable. In the simulated network, the number of CRU pairs and the number of PU pairs are both fixed to five and they are within the transmission range of each other. Each of the PU pairs is assigned a fixed and distinct data channel. More detailed setup specific to a particular network topology will be described in the later subsections.

In all the simulations, we considered an 802.11 CRN with a control channel and five data channels. Each reported performance is the average of ten runs using different random number seeds. For each run, the simulated time is 100 seconds.

4.3 Numerical Evaluation of the BBi-MAC

In the performance study of BBi-MAC, three network scenarios are considered. Firstly, we investigated a simple CRN without the presence of PUs' traffic. In this network

Table 4.3: Simulation Parameters

Parameter Name	**Value**
Number of Data Channel (N_c)	5
Channel Bandwidth	2 Mbps
Channel Sensing Period	2000 us
Fast Channel Sensing	100 us
DIFS	10 us
SIFS	10 us
Duration of QP	100 us
Txop to be used in Section 4.3	1(*), 2-5
Txop to be used in Section 4.4 for Uni-MAC	1-5
MAX_PACKET to be used in Section 4.4 for ABi-MAC	2, 4, 6, 8, 10
Number of PU Pairs	5
Number of CRU Pairs	5

scenario, we first studied the performance gain of BBi-MAC as compared to Uni-MAC when only one CRU pair is generating traffic flow. The numerical results obtained in this network case are used to verify the analytical results as presented in Section 4.1.

Next, we studied the performance of BBi-MAC when the five CRU pairs in the CRN are all generating traffic flows. In this case, all the available data channels are occupied by the CRU pairs alternately. The objective of this evaluation is to study the impact of intra-nodes (i.e., between the CRU pairs) bandwidth contention towards the performance gains of BBi-MAC.

Follow on, we considered a CRN with the presence of PUs' traffic. In this network scenario, five CRU pairs are coexisting with five PU pairs and at least a traffic flow is generated by every node pair. The goal of this evaluation is to study the performance of BBi-MAC in the existence of both the inter-nodes bandwidth contention and intra-nodes (i.e., between the CRU pairs and PU pairs) bandwidth contention.

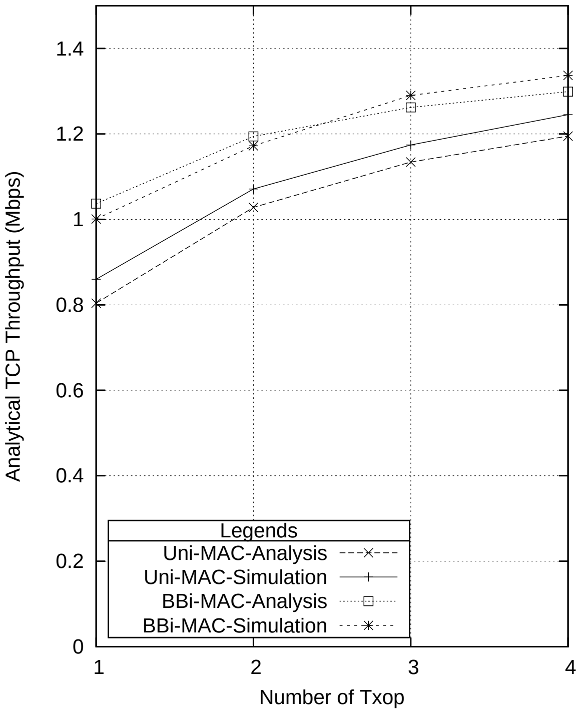


Figure 4.3: Analytical and Simulated TCP Throughputs of Uni-MAC and BBi-MAC

4.3.1 Simulation Results Without the Presence of PUs Traffic

We first studied the throughput performance of the Uni-MAC and BBi-MAC when PUs' traffic flows are absent. In this case, the PUs on the data channels are not generating traffic flows. Instead, each of the CRU pairs generated either a greedy TCP flow or bi-directional greedy UDP flows with fixed 1450-byte packets.

Network Scenario with only a pair of CRUs

In this section, we compared the simulated and analytical throughput results when only one CRU pair is generating traffic flow(s) in the CRN. Fig. 4.3 shows the CRU's analytical and simulated throughput when a TCP flow is being generated. From the figure, one can see that the presented analytical TCP throughputs are slightly less than the simulated TCP throughputs most of the time for both the investigated MAC protocols. This minor deviation is mainly caused by our assumption that every TCP data has to be acknowledged by the receiver in order to simplify our analytical model.

However, in the real world TCP congestion control algorithms, it is not necessarily for a TCP ACK to be replied upon every received TCP data. Particularly, in our simulation environment, the Cubic TCP algorithm is used as default by the Linux machine. By ana-

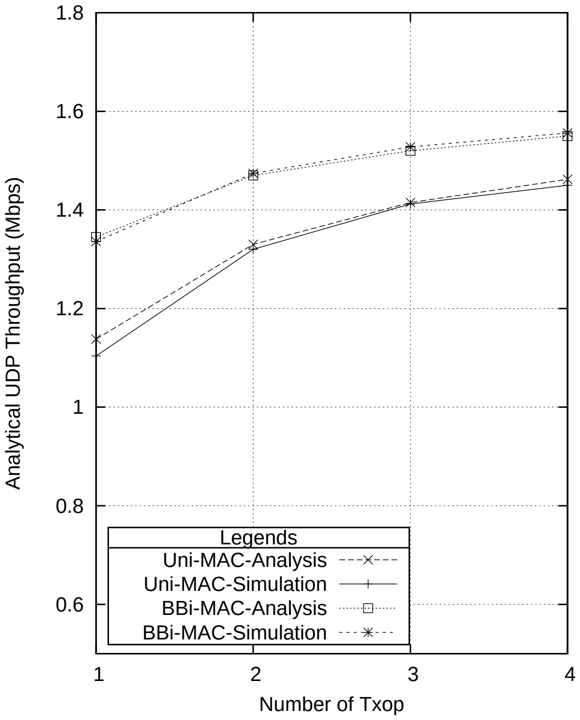


Figure 4.4: Analytical and Simulated UDP Throughputs of Uni-MAC and BBi-MAC

lyzing the captured packets of our simulations through the Wireshark Network Analyzer application program, it is noticed that after the TCP 's congestion window grew to its maximum size, an ACK is being transmitted by the TCP receiver only upon the reception of every two TCP data. Therefore, the simulated TCP throughput is slightly higher than the analytical one as the prior used the bandwidth more effectively in sending useful TCP data.

Such a deviation, however, is not noticed in Fig. 4.4 which shows the aggregate UDP throughput of CRUs when bi-directional UDP flows are being generated. This is because UDP flows do not employ any congestion control mechanism and hence, the flows' throughput are much easier to be estimated analytically. Therefore, the analytical results of UDP throughput match well with the simulated one.

In summary, both of these figures have demonstrated that the analytical results are well verified by our simulation results for both the traffic types. Therefore, it is proven both analytically and numerically that our proposed BBi-MAC performs better than the Uni-MAC, particularly in the presence of bi-directional packets exchange between a CRU pair.

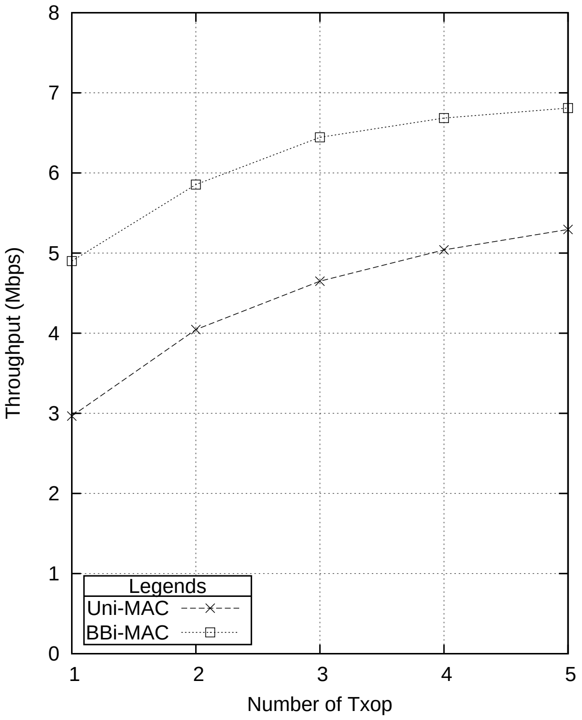


Figure 4.5: Aggregate TCP throughputs of CRUs over Txops (without PUs' traffic)

Network Scenario with Five pairs of CRUs

In this section, we present the simulated throughput results when all the CRU pairs are generating traffic flows in the CRN. Firstly, Fig. 4.5 shows the aggregate TCP throughput of CRUs over *Txop* in the absence of PUs traffic. Again, it is proven from this figure that the BBi-MAC outperforms the Uni-MAC for all the considered *Txops* even in the existence of intra-nodes bandwidth contention. This is because the BBi-MAC has effectively relieved the TCP ACK compression problem by reserving reverse bandwidth for the TCP receiver when a TCP flow is anticipated. With the reserved bandwidth in reverse direction, the transmission rate of TCP ACKs are regulated and the RTT values could be stably maintained at a low value. This results in a significant TCP throughput enhancement.

Another finding from this study is that the TCP throughput improvement achieved the peak (65.26%) when the *Txop* is set to one, and the improvement decreased to 28.61% when the *Txop* is set to five. This is because in the Uni-MAC, when the *Txop* is set to a smaller value, e.g., one, the TCP ACK could be replied quickly in the best case, but may be replied only after a long time in the worst case. Such an uncertainty not only increases the RTT values, but also causes the RTT values to fluctuate in a large range.

Both of these factors lead to a slower growth of the TCP congestion window and thus a low TCP throughput.

In contrast, when the *Txop* is set to a larger value (e.g., five), whenever the TCP receiver successfully gains access to the data channel, it could deliver at most five back-to-back TCP ACKs continuously. These consecutive TCP ACK transmissions will alleviate the fluctuation of RTT values for a TCP flow, thus mitigating the adverse impacts of *ack-compression*. This explains why the throughput improvement of BBi-MAC decreases as the *Txop* value increases.

We also investigated the performance of the two evaluated CR MAC protocols in supporting bi-directional UDP flows. This study is important because the UDP protocol is commonly used in popular applications (e.g., VoIP or video conferencing) over the internet. Fig. 4.6 shows the aggregate UDP throughput of CRUs over *Txops* in the absence of PUs' traffic. The performance trend under UDP is similar to that under TCP. However, because UDP flows are not sensitive to RTT, the UDP flows throughput improved by BBi-MAC comes from the reduced control overheads. Using BBi-MAC, bi-directional data exchange can be arranged in one bandwidth negotiation procedure. In contrast, using Uni-MAC the handshake of bi-directional data exchange requires two bandwidth negotiation procedures to complete. For this reason, BBi-MAC still significantly outperforms Uni-MAC on UDP flow throughputs by reducing its protocol overheads.

Furthermore, we observed that the TCP flows on average achieved a lower aggregate throughput as compared to UDP flows. This is because UDP flows always make full use of the reserved two-way bandwidth for useful packet transmission. However, with TCP flows, the reverse bandwidth is used to carry only the transport layer ACKs which does not add credit to the application layer's throughput. Also, due to the congestion control, TCP flows would control the packet transmission rate based on the RTT and therefore cannot well utilize the link bandwidth. Nevertheless, the results of this study indicate that the proposed BBi-MAC is effective in supporting any kind of traffic flow, regardless whether it is uni-directional or bi-directional.

4.3.2 Simulation Results In the Presence of PUs' Traffic

Next, we studied the performance of the BBi-MAC and Uni-MAC in the presence of PUs' traffic. Firstly, we considered a network scenario where the PUs are generating fixed

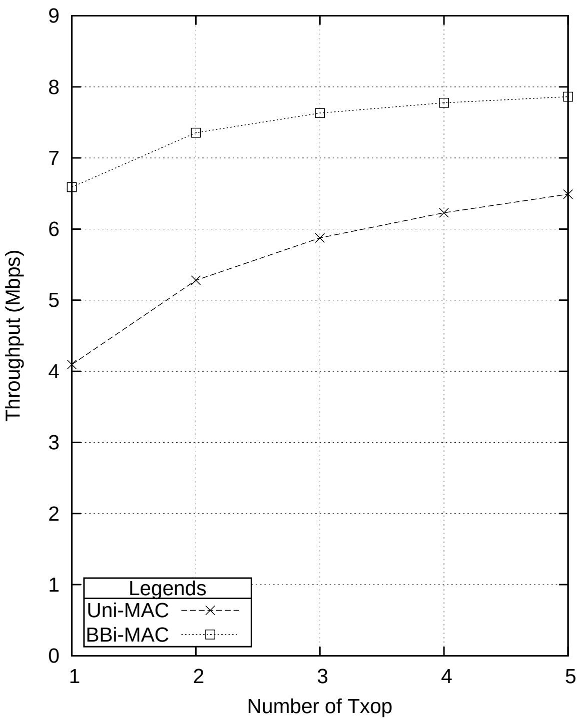


Figure 4.6: Aggregate UDP throughputs of CRUs over Txops (without PUs' traffic)

traffic load. In this case, each of the PU pair generated a UDP flow such that the flow is alternating between ON and OFF periods and consumed an average bandwidth of 0.6 Mbps during every ON periods. Here, we defined the PU load as the average consumed bandwidth when the PUs are transmitting UDP packets during the ON periods. Each CRUs pair generated either a greedy TCP flow or bi-directional UDP flows with a fixed packet size of 500 bytes.

Fig 4.7 and Fig 4.8 show the aggregate TCP throughputs of CRUs over *Txops* and aggregate UDP throughputs of CRUs over *Txops*, respectively. In the presence of PUs' traffic, it is reasonable that the aggregate throughputs of CRUs degraded as the transmission priority is always given to the PUs in a CRN. Therefore, one can see that the performance improvement of BBi-MAC is not as much as that in Section 4.3.1. Nevertheless, the results show that even when PUs' traffic is present, the BBi-MAC can still outperform the Uni-MAC in a significant way. In particular, a maximum improvement of 33.25% in TCP throughput over the BBi-MAC is observed when the *Txop* is set to one. Even when the *Txop* is set to a large value of five, the BBi-MAC still outperformed the Uni-MAC by 20%. One may see that the throughput performance of TCP performance in this section demonstrated a similar trend as that in Section 4.3.1. Therefore we do not repeat the reason here for brevity.

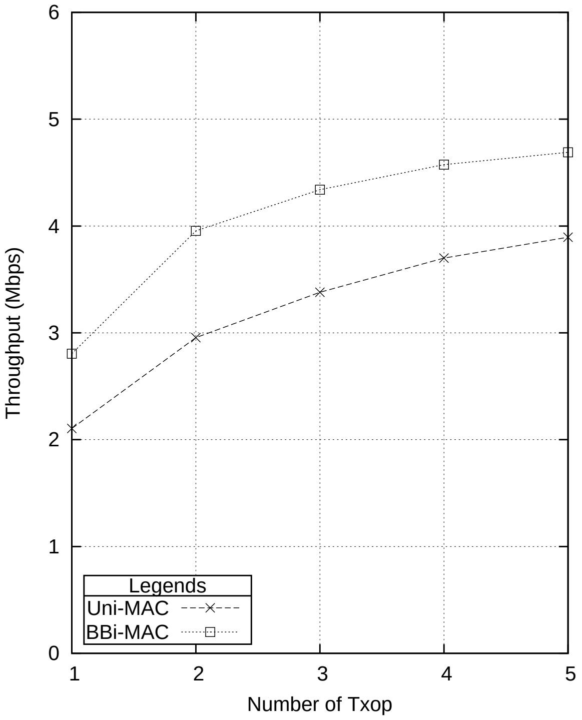


Figure 4.7: Aggregate TCP throughputs of CRUs over *Txops* (with PUs' traffic)

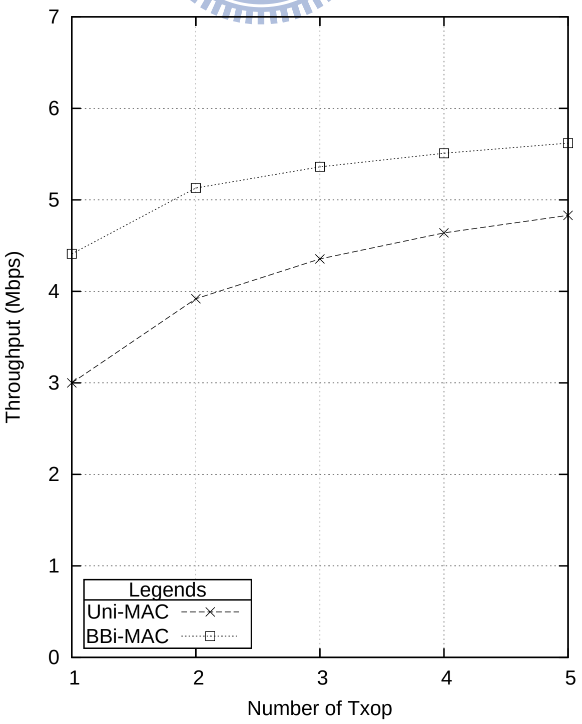


Figure 4.8: Aggregate UDP throughputs of CRUs over *Txops* (with PUs' traffic)

Secondly, we evaluated the TCP performance of CRUs under varying PU loads. We fixed the *Txop* to one and varied the PU load to 0.6 Mbps, 1.2 Mbps and 1.5 Mbps, respectively. Note that due to the extra overhead incurred during data transmission, 1.5 Mbps is considered as the maximum achievable throughput in an 802.11 network with a data channel of 2 Mbps.

Fig 4.9 shows the aggregate TCP throughput of CRUs over different PU loads. As shown in the figure, the TCP throughput over both the considered protocols decreased as the PU load increases. When the PU load is light, the BBi-MAC outperforms the Uni-MAC in a significant way. The improvement decreased as the PU load increases. This is because when the PU load increases, most of the bandwidth is consumed by the PUs and thus the left idle bandwidth has been greatly minimized. In this situation, even the CRUs over the BBi-MAC cannot successfully grab the bandwidth upon packet arrival and therefore obtained a lower throughput. This is, however, a pleasant observation as it shows that both the BBi-MAC and Uni-MAC behave well according to the PU load, which meets the core design requirement of a typical CR MAC protocol.

In addition to investigating the aggregate throughput of CRUs, we also studied whether the additionally reserved bandwidth (in reverse direction) of the BBi-MAC can significantly affect the throughput of PUs. In [10], the authors have shown through simulations that the Uni-MAC could well protect the PUs. Therefore, we used the Uni-MAC as a benchmark here to study the performance of BBi-MAC. Fig 4.10 shows the aggregate throughputs of PUs using both the CR MAC protocols. The figure clearly illustrates that the PUs using the BBi-MAC performed as well as those over the Uni-MAC under all the considered PU loads. Taken together, the results demonstrated in Fig 4.9 and Fig 4.10 suggested that the proposed BBi-MAC significantly outperforms the Uni-MAC and could further boost up the TCP throughput of CRUs without degrading the performance of the PUs.

4.3.3 Summary

The summary of performance improvement through BBi-MAC for the TCP throughput and UDP throughput are presented in Table 4.4 and Table 4.4, respectively. Through mathematical analysis and extensive simulations, it is proven quantitatively that the proposed BBi-MAC protocol can dramatically improve both the TCP and UDP throughput

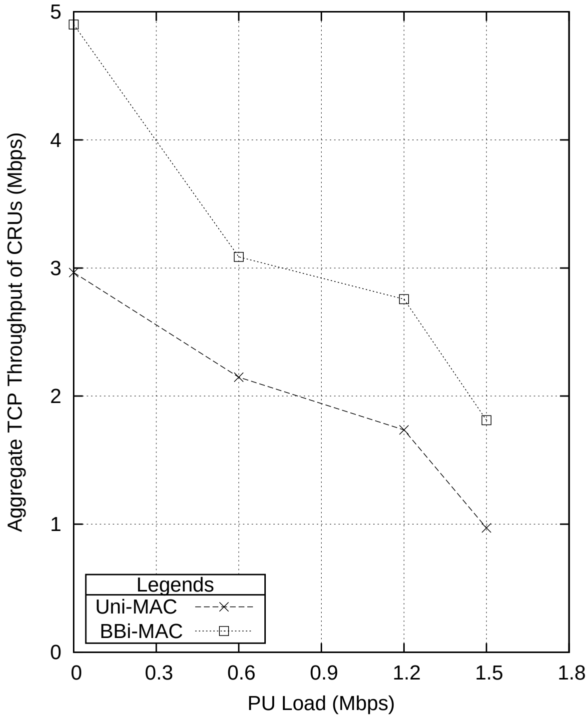


Figure 4.9: Aggregate TCP throughputs of CRUs over PU Loads

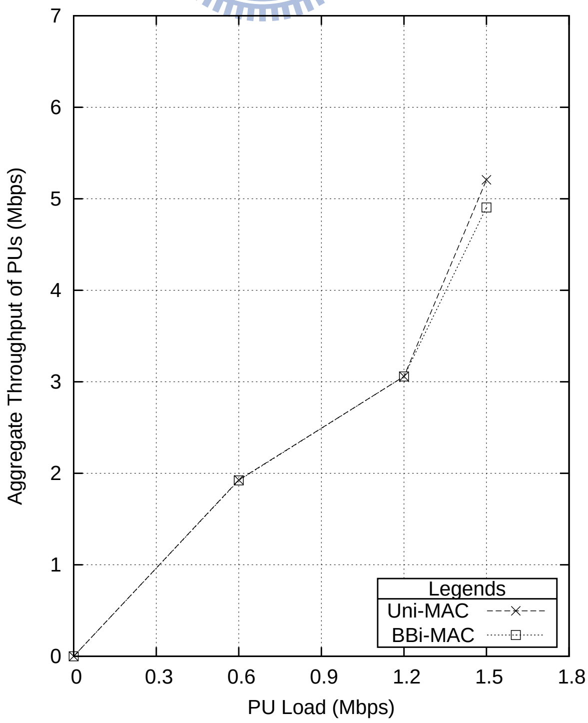


Figure 4.10: Aggregate throughputs of PUs over PU Loads

Table 4.4: Percentage of TCP Throughput Improvement of BBi-MAC as compared to Uni-MAC

Txop	**1**	**2**	**3**	**4**	**5**
1 pair of CRUs	19.69%	9.51%	11.59%	8.26%	6.49%
5 pairs of CRUs	65.26%	44.75%	38.60%	32.64%	28.61%
5 pairs of CRUs + 5 pairs of PUs	33.25%	33.84%	22.84%	23.65%	20.41%

Table 4.5: Percentage of UDP Throughput Improvement of BBi-MAC as compared to Uni-MAC

Txop	**1**	**2**	**3**	**4**	**5**
1 pair of CRUs	21.20%	11.98%	8.36%	6.43%	5.15%
5 pairs of CRUs	60.93%	39.30%	29.87%	24.80%	21.11%
5 pairs of CRUs + 5 pairs of PUs	47.00%	30.87%	23.08%	18.75%	16.36%

of CRUs without sacrificing the performance of PUs.

4.4 Numerical Evaluation of the ABi-MAC

In this section, we evaluated the performance of ABi-MAC through simulations. In the performance study of ABi-MAC, only complex network scenarios are considered. This is because the ABi-MAC is an extension of BBi-MAC, therefore we omitted the simulation cases that have been done for evaluating the performance of BBi-MAC. Generally, network scenarios with five pairs of active CRUs and five pairs of active PUs are considered. However, instead of single TCP flow or two bi-directional UDP flows, here we studied the capability of ABi-MAC when asymmetry traffics are being carried by all the CRU pairs in the CRN.

The two major goals of this investigation are: 1) To explore the impact of *MAX_PACKET* value on the performance of ABi-MAC, and 2) To find the optimal value of *MAX_PACKET* such that it is applicable to all network scenarios by optimizing the trade-off between high throughput of CRUs and low interference to PUs.

4.4.1 Simulation Settings

The simulation settings for the performance evaluation of ABi-MAC is identical to that as shown in Table 4.3. In ABi-MAC, we defined a new variant of *Txop*, namely *MAX_PACKET*, which refers to the maximum number of packets that is allowable to be transmitted by a CRU pair when both are on the data channel. Since the definition of *MAX_PACKET* in ABi-MAC is different from that of *Txop* in Uni-MAC, it is illogical to compare these two protocols directly by taking *MAX_PACKET*=*Txop*. Therefore, based on the philosophy that every even *MAX_PACKET* (i.e., two, four, six, etc.) setting of ABi-MAC may carry the same amount of packets as BBi-MAC if the *Txop* of BBi-MAC is being set to one, two , three, etc., we defined the following five comparison cases, aiming to make the comparison between the ABi-MAC and Uni-MAC more consistent and fairer:

- Case 1: Comparison between Uni-MAC with one *Txop* (Uni(1)) and ABi-MAC with two *MAX_PACKET* (ABi(2)).
- Case 2: Comparison between Uni-MAC with two *Txop* (Uni(2)) and ABi-MAC with four *MAX_PACKET* (ABi(4)).
- Case 3: Comparison between Uni-MAC with three *Txop* (Uni(3)) and ABi-MAC with six *MAX_PACKET* (ABi(6)).
- Case 4: Comparison between Uni-MAC with four *Txop* (Uni(4)) and ABi-MAC with eight *MAX_PACKET* (ABi(8)).
- Case 5: Comparison between Uni-MAC with five *Txop* (Uni(5)) and ABi-MAC with ten *MAX_PACKET* (ABi(10)).

It is worth noting that due to the consideration that CRUs are secondary users of the CRN, we do not want the CRUs to monopoly a data channel by spending long duration on that channel. This is because by staying too long on a data channel, the CRUs will affect the PUs' performances especially when the network is busy. In addition, it will result in unfairness issue such that some other CRUs may not have any opportunities to utilize the available spectrum holes. Therefore, we intentionally limited the *Txop* of Uni-MAC to a value of five, and the *MAX_PACKET* of ABi-MAC to a value of ten.

In the simulated networks, asymmetry traffics are being generated by every CRU pairs. The asymmetry traffics are composed of:

- One-way TCP flow from the CR sender to CR receiver: The TCP flow is alternating between ON and OFF periods throughout the 100 seconds simulation time.

- One-way UDP flow from the CR receiver to CR sender: The UDP flow is alternating between ON and OFF periods throughout the 100 seconds simulation time. During the ON periods, the UDP packets arrive at a constant inter arrival time of 0.0005 seconds and the packet sizes follow an exponential distribution with a mean packet size of 900 Bytes.

In addition to studying the impact of *MAX_PACKET* on the protocol's performance, we also investigated how the PUs' traffic may affect the achievable CRUs throughputs. To this end, we defined the 'PU load' as the ratio of aggregate PUs throughput over the maximum achievable throughput of PUs in the CRN . In the simulated cases, each of the PU pair generated either a UDP flow or two bi-directional UDP flows such that each of the flow is alternating between ON and OFF periods and consumed a fixed amount of average bandwidth during every ON periods. The aggregate PUs throughput is thus defined as the aggregate throughputs of PUs when these PUs are transmitting UDP packets during the ON periods. The three different cases of PU load that are taken into consideration in our simulations are 0.30, 0.50 and 0.80, which are referred to as light PU load, medium PU load and heavy PU load, respectively.

4.4.2 Simulation Results of ABi-MAC

In this section, we present the simulated throughput results when all the CRU pairs are generating asymmetry traffic flows in the CRN. Firstly, Fig. 4.11 shows the aggregate throughputs of CRUs operating using the Uni-MAC and ABi-MAC, respectively, when the PU load is light. As shown in the figure, the ABi-MAC outperforms the Uni-MAC for all the considered comparison cases. This significant improvement is mainly due to the design of ABi-MAC which can adaptively reserve bandwidth to either the CR sender or CR receiver or both depends on the CRU's instantaneous traffic needs. Such an adaptive bandwidth allocation makes sure that spectrum holes can be optimized by giving chances only to CRUs who have packets to be transmitted.

Similar as those observations in Section 4.3, one can see that the throughput improvement of ABi-MAC achieved the peak (42%) in case one, and its throughput improvement

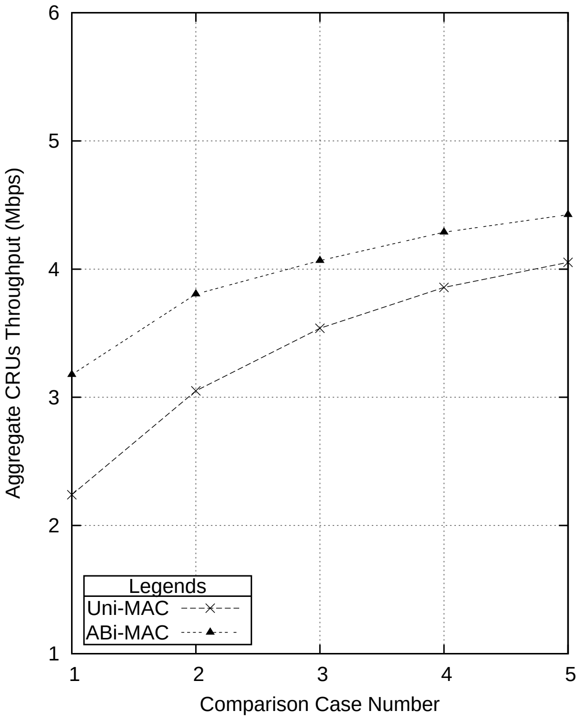


Figure 4.11: Aggregate throughputs of CRUs when PU load is light

decreased to 10% in case five. Since the aggregate CRUs of Fig. 4.11 shows a similar trend as that in Section 4.3, we do not repeat the reason here for brevity. The same performance trends are also noted in Fig. 4.12 and Fig. 4.13, which show the aggregate throughputs of CRUs when the PU load is medium and high, respectively.

However, different from the simulations in Section 4.3, asymmetry traffic flows are considered in the simulations in this section. Therefore, the performance gain of ABi-MAC as shown in Fig. 4.11, Fig. 4.12 and Fig. 4.13 is caused by the combination factors of reduced RTT of TCP flow and reduced protocol overhead in transmitting the UDP flow. Moreover, from the summary of throughput improvement as shown in Tab. 4.6, it is apparent that when PU load in the network increases, the throughput improvement achieved by ABi-MAC also increases. For example, referring to the case one as displayed in the table, one can see that the throughput improvement of ABi-MAC increases from 42% to 62%, lastly to 78%, when the PU load in the network is increased from 0.3 to 0.5, and lastly to 0.8. All of these statistics suggests that the ABi-MAC always outperforms the Uni-MAC regardless of the network condition (i.e., idle or busy) and the performance enhancement of ABi-MAC is getting more significant when the considered CRN consists of busier PUs.

By comparing the aggregate throughputs of Fig. 4.11, Fig. 4.12 and Fig. 4.13, one

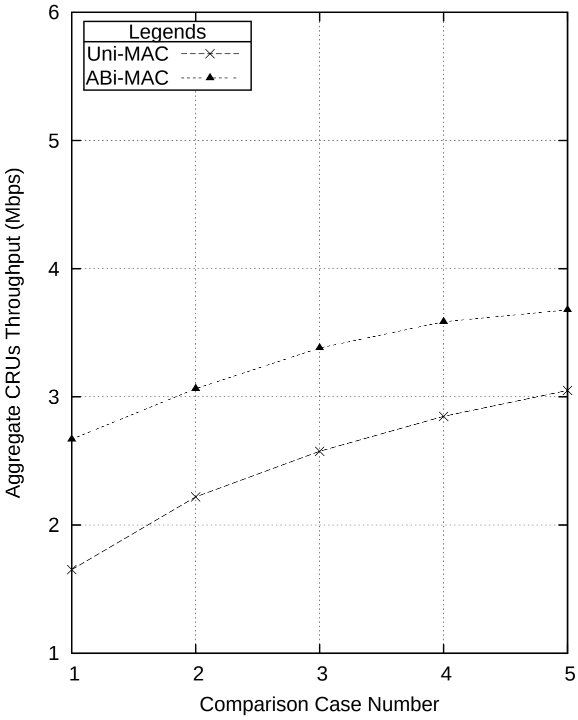


Figure 4.12: Aggregate throughputs of CRUs when PU load is medium

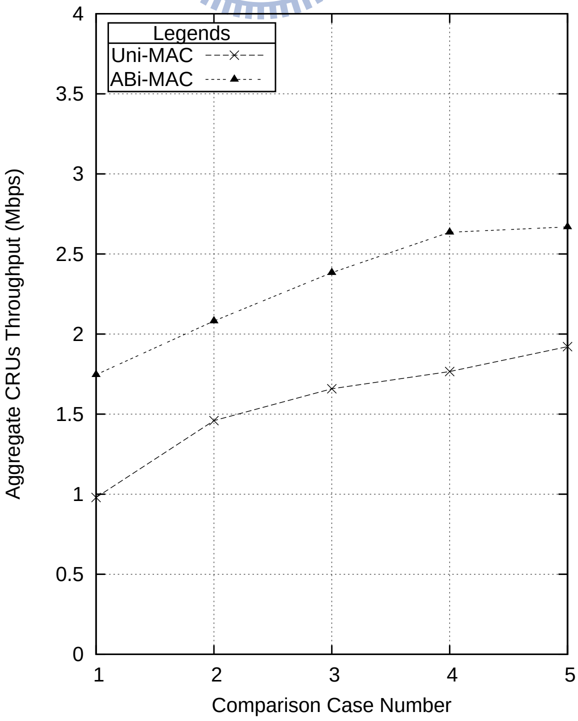


Figure 4.13: Aggregate throughputs of CRUs when PU load is high

Table 4.6: Summary of Throughput Improvement of ABi-MAC as compared to Uni-MAC

Case Number	**1**	**2**	**3**	**4**	**5**
PU load - Light	42%	25%	14%	11%	10%
PU load - Medium	62%	38%	30%	25%	20%
PU load - High	78%	43%	43%	42%	39%

can notice that the aggregate CRUs throughputs of ABi-MAC decreases as the PU load is increasing from light load to medium load, and eventually high load. Though the ABi-MAC may reserve more bandwidth to CRUs in the presence of bi-directional packet exchanges at a CRU pair, the above observation however proves that ABi-MAC still offers good protection to PUs by giving them the top priority to use the data channel. Therefore, when the PU load increases, the CRUs have less spectrum holes to transmit their packets, thus decreasing the aggregate throughputs.

In all the considered cases as shown in the figures, we also observed that the aggregate CRUs throughput increases as the *MAX_PACKET* increases. However, the throughput improvement becomes less significant when the *MAX_PACKET* is increased from a value of eight to a value of ten. To be specific, throughput improvements of only 1.2%, 2.5% and 3.2% are noted when *MAX_PACKET* is increased from eight to ten in network scenarios with light PU load, medium PU load and high PU load, respectively. Such an observation suggests that a large value of *MAX_PACKET* (i.e., ten) does not necessary improve the protocol's performance in an impressive way. Therefore, we conclude that *MAX_PACKET* with a value of eight is an optimal setting for ABi-MAC since it can optimize the aggregate CRU throughputs in all kinds of network scenarios while offering an adequate protection to the performances of PUs.

Follow on, we fixed the *MAX_PACKET* to a value of eight and compared the performance of ABi-MAC (denoted as ABi(8)) against Uni-MAC with different *Txop* settings (denoted as Uni(1), Uni(2), Uni(3), Uni(4), and Uni(5), respectively). Fig. 4.14 shows the throughput comparison of ABi-MAC and Uni-MAC under network scenarios with different PU loads. From the figure, one can see that for the three considered network scenarios with different PU loads, the ABi-MAC always outperforms the Uni-MAC regardless of the *Txop* settings. Even when the Uni-MAC operated using the maximum

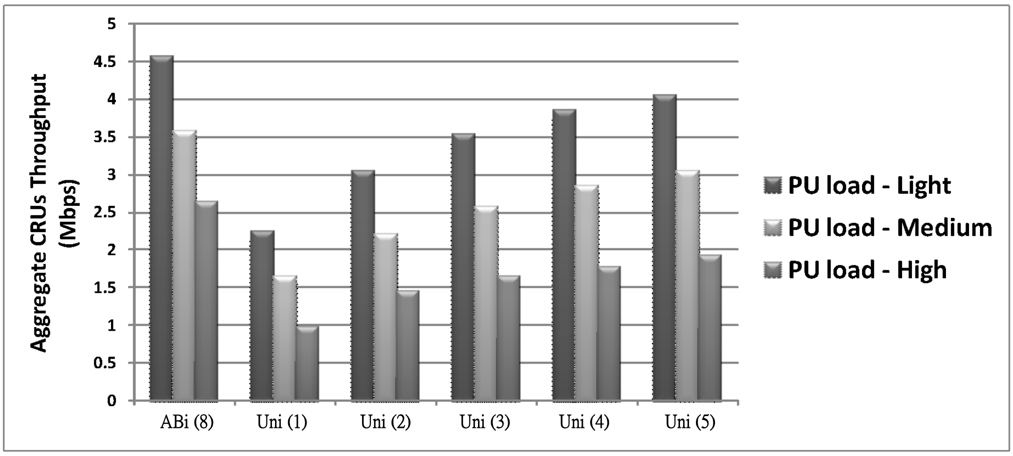


Figure 4.14: Aggregate CRUs throughputs of ABi-MAC and Uni-MAC under different PU loads

value of *Txop* (i.e., Uni(5)), the achievable aggregate throughput is still less than than of ABi(8). Moreover, it is noticed that the aggregate throughput difference between the ABi(8) and Uni(5) becomes larger when PU load increases. All of these observations further suggest that setting the *MAX_PACKET* of ABi-MAC to a value of eight is an optimum network parameter setting which can optimize the CRN performance in all kinds of network scenarios.

Lastly, we studied the impact of ABi-MAC towards the PUs' performances. In Section 4.3.2, we have proven that the PUs are well protected in terms of their throughputs performance when BBi-MAC is adopted. In this section, instead of PUs' throughputs, we defined "maximum transmission interval (MTI) of PUs" as a new PUs' performance metric. The MTI refers to the maximum waiting time of a PU in between the transmission of two consecutive data packets. In brief, the MTI shall strictly include the time that a PU sender senses the medium as idle and successfully exchanging RTS-CTS control packets with a PU receiver.

Fig. 4.15 shows the average MTI of PUs when they are coexisting with CRUs under all considered cases of PU loads. As shown in the figure, one can see that in the absence of CRUs, the average MTI of PUs is stably maintained around 30ms for all the comparison cases. When CRUs operating using Uni-MAC are present in the CRN, the MTI of PUs started to increase such that the larger the adopted value of *Txop*, the longer the MTI. In the consideration of CRUs operating using ABi-MAC, the MTI of PUs is further boosted up such that it is almost doubled as compared to the case when CRUs operating using

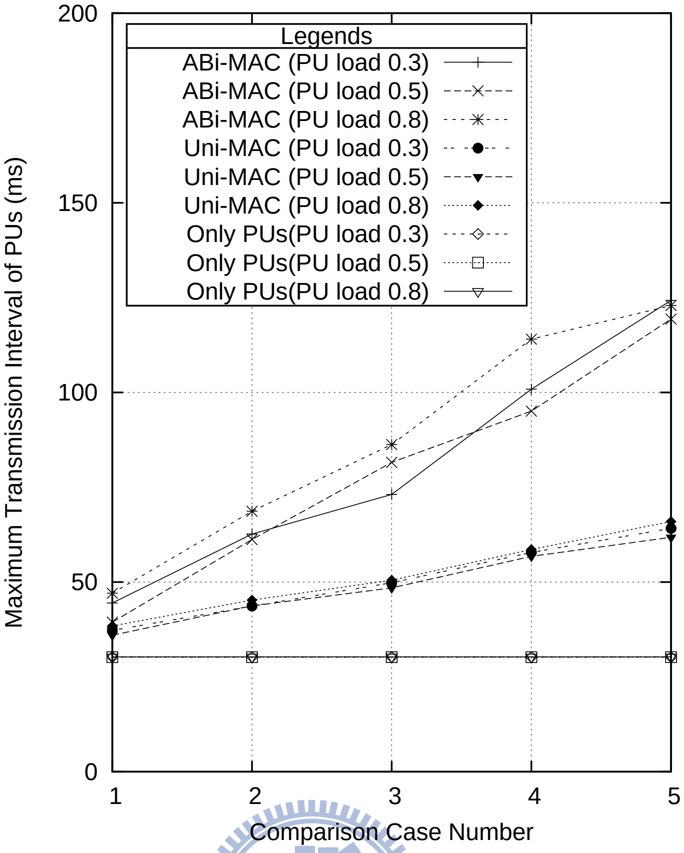


Figure 4.15: Maximum Transmission Interval of PUs

Uni-MAC are present. Such an observation is reasonable because in the worst case, the CRU pair under ABi-MAC may exchange two large data packets of almost equal size, causing the MTI of PUs being doubled.

Nevertheless, one can see that even in the comparison case five, the average MTI of PUs is well kept below 150 ms, which means that the PUs are only being affected in a very minor and insignificant way which is tolerable by the PUs. In fact, in all the considered cases (i.e., PUs without the absence of CRUs, PUs togetehr with CRUs operating using Uni-MAC and PUs together with CRUs operating using ABi-MAC) under different PU loads, the average waiting interval of PUs is stably maintained around 12.5 ms. This further proves that in most of the time, the CRUs are using the spectrum for packet transmissions only when PUs are idle, therefore the average waiting interval of PUs in the cases of with and without CRUs does not show any difference.

4.4.3 Summary

In summary, the presented simulation results in this section have proven that ABi-MAC still outperforms the Uni-MAC even when complex network scenarios are considered. Furthermore, it is shown that the performance enhancement of ABi-MAC is getting more significant in the presence of heavier PU traffics. In addition, we also identified that ABi-MAC with its *MAX_PACKET* being set to a value of eight is an optimum protocol which can always optimize the CRN performance in all kinds of network scenarios.

Chapter 5

Conclusion

5.1 Final Remarks

In this dissertation, we point out the deficiencies of two major related works and propose two schemes of CR MAC protocols to enhance the bandwidth efficiency of a CRN. Firstly, we propose a basic scheme of bi-directional CR MAC protocol, namely BBi-MAC, to better support traffic flows with bi-directional packets exchanges, particularly the TCP flows and bi-directional UDP flows. Based on BBi-MAC, we further expand the protocol and propose an advanced scheme of bi-directional CR MAC protocol, namely ABi-MAC. The proposed ABi-MAC possesses the ability of reserving the bandwidth dynamically to further well utilize the spectrum holes of the authorized bandwidth. In addition, the ABi-MAC can support asymmetry two-way traffic flows with variable packet sizes. The performances of our proposed schemes are evaluated using both mathematical analysis and simulations. Results from both of the adopted evaluation approaches proved that our proposed schemes can significantly enhance the aggregate throughputs of CRUs in a CRN and dramatically improve the spectrum efficiency without degrading the performance of PUs.

5.2 Future Work

The proposed CR MAC protocols can be further extended from several aspects, which are discussed below.

5.2.1 Cognitive Mesh Networks

The main contribution of our proposed protocols is to enhance the spectrum efficiency by increasing the throughputs between a pair of CRUs such that the CR sender communicates with the CR receiver through one-hop ad hoc connection on the unlicensed 802.11 spectrum band. Although our protocols can also work in a mesh (i.e., multi-hop) environment, the protocol's performance is expected to degrade drastically. This is because as compared to the classical wireless mesh networks, cognitive mesh networks possess the following unique features and challenges:

- Due to the PUs' activities, the data channels are not continuously available for packets transmission. Therefore, each CRU shows different link availability which usually vary rapidly across time. As a result, in a cognitive mesh networks, it is very likely that the end-to-end route consisting of multiple hops would have different channels according to the spectrum availability. Moreover, the spectrum switches on the links are frequent based on PU arrivals. Therefore, it would be an important research to integrate the routing and spectrum allocation in establishing these end-to-end routes.

- The ad hoc cognitive network lacks centralized support, and hence must reply on local coordination to gather topology information. In classical wireless mesh network, this is easily accomplished by periodic beacon messages on the channel. However, in cognitive mesh networks, sending beacons over all the data channels is not feasible because these beacons may interfere with PUs' activity. Furthermore, the CRUs may switch between channels (i.e., control channel and data channels) from time to time, therefore sending beacons over the control channel is not feasible too because the CRUs are not listening to the control channel all the time. Therefore, cognitive mesh networks are highly probable to have incomplete topology information, which increases the difficulties in establishing an optimal end-to-end route.

- In classical mesh networks, route formed over multiple hops may periodically experience disconnections caused by node mobility. These cases may be detected when the next hop node in the path does not reply to messages and the retry limit is exceeded at the link layer. However, in cognitive mesh network, in addition to node mobility, a node may not be able to transmit immediately if it detects the presence of a PU on the spectrum. Thus, correctly inferring mobility conditions and initiating the appropriate recovery mechanism in cognitive mesh networks necessitate a different approach from the classical ad hoc networks.

Due to the challenges mentioned above, more research is needed for CR MAC protocols which can support cognitive mesh networks.

5.2.2 Cognitive Heterogeneous Networks

Our CR MAC protocol can coexist with the IEEE 802.11 networks. However, it is envisaged that cognitive heterogeneous networks will be a future trend of wireless world. In this case, in addition to the IEEE 802.11 WLAN, the CR MAC protocol should coexist with some others access networks, such as IEEE 802.15 WPAN, IEEE 802.16 WMAN, UMTS and etc. Therefore, it would be another challenging research to expand or even modify our CR MAC protocol such that it can incorporate with the future cognitive heterogeneous networks.

Bibliography

[1] Federal Communications Commision, Spectrum Policy Task Force, *Report of the Spectrum Efficiency Working Group*, Nov15, 2002.

[2] V. Valenta, Z. Fedra, R. Marsalek, G. Baudoin, and M. Villegas, “Towards Cognitive Radio Networks: Spectrum Utilization Measurements in Suburb Environment,” in *Radio and Wireless Symposium*, San Diego, CA, USA, 18-22 Jan 2009.

[3] Mitola,J., III and Maguire, G.Q., Jr., “Cognitive Radio: Making Software Radios more Personal,” *IEEE Transactions on Personal Communications*, vol. 6, no. 4, August 1999.

[4] Mitola,J., III, “Cognitive Radio for Flexible Mobile Multimedia Communications,” in *IEEE International Workshop on Mobile Multimedia Communications (MoMuC)*, San Diego, CA, USA, 15-17 Nov 1999.

[5] Mitola,J., “The Software Radio,” in *IEEE National Telesystems Conference*, 1992.

[6] IEEE 802.22 Working Group, *Wireless Regional Area Networks Enabling Rural Broadband Wireless Access.*

[7] Standard ECMA-392, *MAC and PHY for Operation in TV White Space*, Dec 2009.

[8] Yingxi Liu and N. Kundargi, “A Novel Sense-Transmit-Wait Strategy for Coexistence of Cognitive Radio Networks with IEEE 802.11 WLANs,” in *The International Symposium on Communications, Control and Signal Processing 2010 (ISCCSP 2010)*, Limassol, Cyprus, 3-5 March 2010.

[9] 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.

[10] L. Shie-Yuan Wang, Yu-Ming Huang and Chih-Che Lin, “Enhanced MAC Protocol for Cognitive Radios over IEEE 802.11 Networks,” in *IEEE Wireless Communication and Networking Conference 2011 (WCNC 2011)*, Cancun, Mexico, 28-30 March 2011.

[11] D. Niyato and E. Hossain, “Cognitive Radio for Next-Generation Wireless Networks: An Approach to Opportunistic Channel Selection in IEEE 802.11-based Wireless Mesh,” *IEEE Wireless Communications*, pp. 46–54, Feb. 2009.

[12] Pablo Brenner, “A Technical Tutorial on the IEEE 802.11 Protocol,” in *Breezecom Wireless Communications)*, July 1996.

[13] Yingxi-Liu and N. Kundargi, “A Novel Sense-Transmit-Wait Strategy for Coexistence of Cognitive Radio Networks with IEEE 802.11 WLANs,” in *The International Symposium on Communications, Control and Signal Processing 2010 (ISCCSP 2010)*, Limassol, Cyprus, 3-5 March 2010.

[14] W. Shao-Yi Hung, Yu-Chun Cheng and Gen-Huey Chen, “An Opportunistic MAC Protocol for Coexistence with WLAN,” in *IEEE International Conference on Communications 2008 (ICC 2008)*, Beijing, China, 19-23 May 2008.

[15] S. Tanaka and T. Fujii, “Wireless LAN Protected Distributed Cognitive MAC Protocol for Secondary System,” in *The International Workshop on Cognitive Radio and Advanced Spectrum Management 2009 (CogART 2009)*, Aolborg, Denmark, 18-20 May 2009.

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

Publications

- Conference

  1. 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.
  2. **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.
  3. Shie-Yuan Wang, Bo-Fan Wang, Yu-Wei Lee and **Lee-Chin Lau**, "Design and Implementation of a more Realistic Radio Propagation Model for Wireless Vehicular Networks over the NCTUns Network Simulator," IEEE Wireless Communication and Networking Conference 2011 (WCNC 2011), 28-30 March 2011, Cancun, Mexico.

100

碩士論文

在IEEE 802.11網路上具有動態頻寬配置功能的
雙向感知無線電媒介層多重存取協

交通大學
資訊學院
資訊科學與工程研究所

劉莉晶

9855632

100

碩士論文

在IEEE 802.11網路上具有動態頻寬配置功能的
雙向感知無線電媒介層多重存取協

交通大學
資訊學院
資訊科學與工程研究所

劉莉晶

9855632

100

碩士論文

在IEEE 802.11網路上具有動態頻寬配置功能的
雙向感知無線電媒介層多重存取協

交通大學
資訊學院
資訊科學與工程研究所

劉莉晶

9855632

100

碩士論文

在IEEE 802.11網路上具有動態頻寬配置功能的
雙向感知無線電媒介層多重存取協

交通大學
資訊學院
資訊科學與工程研究所

劉莉晶

9855632

100

碩士論文

在IEEE 802.11網路上具有動態頻寬配置功能的
雙向感知無線電媒介層多重存取協

交通大學
資訊學院
資訊科學與工程研究所

劉莉晶

9855632

學號