Ph.D. DISSERTATION

MAC Protocols for Enhancing Network Performance by Adjusting Frame Transmission Duration in IEEE 802.11n WLANs

IEEE 802.11n WLAN에서 프레임 전송 시간 조절을 통한 네트워크 성능 향상 MAC 프로토콜

BY

YOUNGSOO LEE

AUGUST 2013

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Ph.D. DISSERTATION

MAC Protocols for Enhancing Network Performance by Adjusting Frame Transmission Duration in IEEE 802.11n WLANs

IEEE 802.11n WLAN에서 프레임 전송 시간 조절을 통한 네트워크 성능 향상 MAC 프로토콜

BY

YOUNGSOO LEE

AUGUST 2013

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

MAC Protocols for Enhancing Network Performance by Adjusting Frame Transmission Duration in IEEE 802.11n WLANs

IEEE 802.11n WLAN에서 프레임 전송 시간 조절을 통한 네트워크 성능 향상 MAC 프로토콜

지도교수 최종호

이 논문을 공학박사 학위논문으로 제출함

2013 년 7 월

서울대학교 대학원

전기 정보 공학부

이영수

이영수의 공학박사 학위논문을 인준함

2013 년 6 월

위원장	:	박세웅
부위원장	:	최종호
위원	:	최성현
위원	:	오성회
위원	:	박은찬

Abstract

The demand for wireless local area network (WLAN) has drastically increased due to the prevalence of the mobile devices such as smart phones and tablet PCs. However, the distributed coordination function (DCF), which is the basic MAC protocol used in IEEE 802.11 WLANs, needs to be improved on MAC efficiency in single-cell networks and fairness performance in ad-hoc networks. In this dissertation, we propose two MAC protocols that can enhance MAC efficiency in single-cell network, and max-min air-time fairness in ad-hoc network by adjusting frame transmission duration, respectively. In the traditional MAC protocol, the length of a packet or a frame is usually fixed and the transmission duration is determined by the data rate. However, we show how each node can precisely adjust the transmission duration when the frame aggregation and block ACK features are used in very high-speed IEEE 802.11n/ac/ad WLANs. If the transmission duration can be precisely controlled, it plays the role usually carried out by a control message. Therefore, a node can indirectly announce necessary information to the other nodes, which can sense the transmission of the node, without incurring any overhead. This idea is simple, but very effective to enhance the network performance by exchanging the necessary information without overheads.

First, we propose the Transmission Order Deducing MAC (TOD-MAC) protocol to improve MAC layer efficiency in IEEE 802.11 single-cell network. Recently, the physical (PHY) layer transmission rate increases to Gbps range in the IEEE 802.11 WLANs. However, the increase in the PHY layer transmission rates does not necessarily translate into corresponding increase in the MAC layer throughput of IEEE 802.11 WLANs because of MAC overheads such as PHY headers and contention time. TOD-MAC precisely controls the frame length and transmission duration to indirectly provide necessary information to a node to determine the transmission order among all the nodes in a network. Each node transmits frames of different duration, and thus the other nodes can determine the time when they can transmit, which has the same effect as announcing the transmission order, without using a control message. Each node transmits a frame in a round robin manner, which minimizes the idle time between two consecutive transmissions without collisions, and significantly enhances the MAC efficiency in very high speed CSMA/CA wireless networks. The results of extensive simulations indicate that TOD-MAC achieves high throughput performance, short/long-term air-time fairness in multi-rate networks and excellent transient behavior in dynamic environments.

Secondly, we propose Max-min Air-time Fairness MAC (MAF-MAC) to improve max-min air-time fairness in IEEE 802.11 ad-hoc networks. As the demand for services based on ad-hoc networks rapidly increases, enhancing fairness among nodes becomes important issue in ad-hoc networks. The concept of max-min fairness is that a node may use more channel resource as long as it does take away the channel resource from the other nodes who uses less channel resource. In MAF-MAC, the transmission duration is adjusted so that it can indirectly perform the function of a control message in announcing the state of a node, called the busy time ratio. On the basis of this information, each node adjusts its CW value to improve max-min air-time fairness. Moreover, we also adopt the hidden node detection and resolving mechanism to MAF-MAC to improve the max-min air-time fairness even when there are hidden nodes in ad-hoc networks. We show through simulation that MAF-MAC incorporating hidden node detection/resolution mechanisms can provide good air-time fairness with high channel occupation and utilization ratio whether or not there are hidden nodes in the network.

Keywords: Wireless LAN, IEEE 802.11, IEEE 802.11n, medium access control (MAC), round robin, transmission duration adjusting, max-min fairness, air-time fairness, hidden node problem

Student Number: 2010-30225

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The users of wireless mobile devices such as smart phones and tablet PCs, which basically require wireless Internet access, is explosively increased, and the wireless local area network (WLAN) is one of the most popular wireless communication technology thanks to its ease of deployment and low installation cost. At the same time, the demand for multimedia applications such as HDTV (20 Mbps) and DVD (9.8 Mbps) rapidly increases in WLANs, and higher bandwidth is required for such services in wireless networks. To satisfy the increasing demand for higher throughput of WLANs, the IEEE 802.11n standard introduces new physical (PHY) layer and medium access control (MAC) specifications [1]. By using advanced PHY layer technologies such as multiple-input multiple-output (MIMO) antenna, orthogonal frequency division multiplexing (OFDM), adaptive channel coding, and channel bonding, the data rate in the PHY layer reaches up to 600 Mb/s for a 40 MHz channel bandwidth and 4x4 MIMO configuration. Furthermore, IEEE 802.11ac/ad aims to support PHY layer rates in the Gbps range [2, 3]. Although the transmission rate may significantly increase, one does not see a commensurate increase in user throughput because the MAC efficiency of IEEE 802.11 rapidly decreases with increasing PHY rates [4,5]. This is because the MAC-layer overheads such as the MAC header, contention time, and acknowledgement (ACK) frame transmission limit user throughput. In fact, in the transmission of a frame, the proportion attributed to overhead becomes larger as the PHY rate increases. According to a study conducted by $Li\ et\ al.$ [4], MAC efficiency falls from 42% at a 54 Mbps rate to only 10% at a 432 Mbps rate.

IEEE 802.11n [1] introduced several mechanisms including frame aggregation and Block ACK to enhance MAC efficiency. In the traditional MAC protocol, the length of a packet or a frame is usually fixed and the transmission duration is determined by the data rate. However, we will show shortly that a node can precisely adjust the transmission duration when the frame aggregation and block ACK features are used as in very high-speed IEEE 802.11n/ac/ad WLANs. If the transmission duration can be precisely controlled, it can play the role usually carried out by control messages, i.e., a node can indirectly announce necessary information to the other neighbor nodes, which can sense the transmission of the node, without incurring any overhead. By using transmission duration, necessary information can be exchanged among the nodes that are in the carrier sensing range of each other. This is better than transmitting a control message directly or using an optional field in the PHY/MAC headers, because nodes must be in the transmission range of each other to communicate successfully. This dissertation is based on this idea, which is simple, but very effective to enhance network performance by exchanging necessary information without overheads. In this dissertation, we propose two MAC protocols that can enhance MAC efficiency in single-cell network, and max-min air-time fairness in ad-hoc network by adjusting frame transmission duration, respectively. It is noted that we will discuss the operation of the proposed MAC protocols based on IEEE 802.11 for ease of explanation, but its main underlying idea can be applied to any carrier sense multiple access with collision avoidance (CSMA/CA) wireless networks.

In the first part of this dissertation, we propose Transmission Order Deducing MAC (TOD-

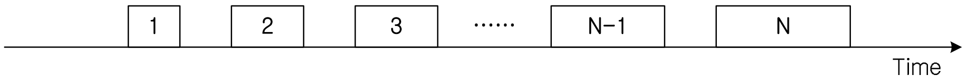


Figure 1.1 Data transmission in a round robin manner

MAC) protocol to improve MAC layer efficiency in IEEE single-cell networks. As the PHY rate increases, the time to transmit a frame is quickly dominated by a fixed overhead associated with the PHY header, contention time, etc. Reducing the wasted time caused by collisions or channel errors is crucial for improving the MAC efficiency. Data transmission in a round robin manner, instead of contention for an opportunity for data transmission, is an attractive alternative. Figure 1.1 shows the basic concept of the round robin data transmission when there are N number of nodes in a single-cell network. If each node transmits in a round robin manner, the both contention time and collision rate can be minimized at the same time, and consequently the MAC efficiency can be improved. TOD-MAC protocol is based on a round robin scheme, but it does not use any control messages. In TOD-MAC, the transmission duration is accurately adjusted and it performs the function of a control message to determine the transmission order of nodes. Based on the information of transmission order, each node transmits in a round robin manner. In this way, TOD-MAC can achieve a very high MAC efficiency in various network environments. Furthermore, the results of extensive simulations show that TOD-MAC provides a good short/long term air-time fairness, and fast transient response in dynamic environments, where a node newly joins or leaves.

As the demand for services based on ad-hoc networks rapidly increases in WLANs, enhancing fairness among nodes becomes an important issue in ad-hoc networks. If each node can share the wireless channel fairly, it will be satisfied. However, sometimes, some nodes monopolize the channel and some nodes cannot use the channel at all due to the relative position of the nodes in a network. Because each node operates based on CSMA/CA in IEEE 802.11 WLANs, a node that senses the channel busy for all the time never has a chance to transmit a

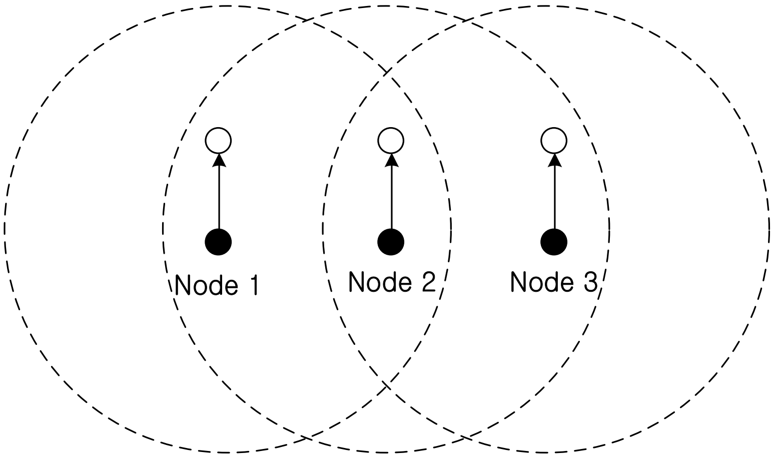


Figure 1.2 The simple starvation scenario in the ad-hoc networks

data frame. In ad-hoc networks, a node generally has different number of nodes in its carrier sensing range from the other nodes. Then, a node that has many nodes in its carrier sensing range may starve due to the nodes in the carrier sensing range. Figure 1.2 shows a simple starvation scenario in an ad-hoc network, where the circles with the dotted-line indicate the carrier sensing range. In this scenario, node 2 senses the channel busy most of time and suffers from starvation because nodes 1 and 3 monopolize the channel. To solve such an unfairness problem, in the second part of this dissertation, we propose a MAC protocol that will improve max-min air-time fairness in IEEE 802.11n ad-hoc networks. It is noted that max-min fairness implies achieving not only the air-time fairness, but also full utilization of the channel. This protocol will be called as Max-min Air-time Fairness MAC (MAF-MAC). In MAF-MAC, a node estimates the ratio of its using air-time with respect to the total channel busy time starting from its present transmission to the next transmission, and announces estimated ratio to the nodes in its carrier sensing range. By using the information of estimated ratio, each node can appropriately adjust its CW value to improve max-min air-time fairness in ad-hoc network. Note that there may be hidden nodes in an ad-hoc network, and the fairness can be significantly degraded by hidden nodes [6, 7]. Therefore, we adopt the hidden node detection [8] and resolving mechanism [9] to MAF-MAC for alleviating the hidden node problem. We show in a simulation study that MAF-MAC improves air-time fairness and at the same time fully and effectively utilize the channel whether or not there are hidden nodes.

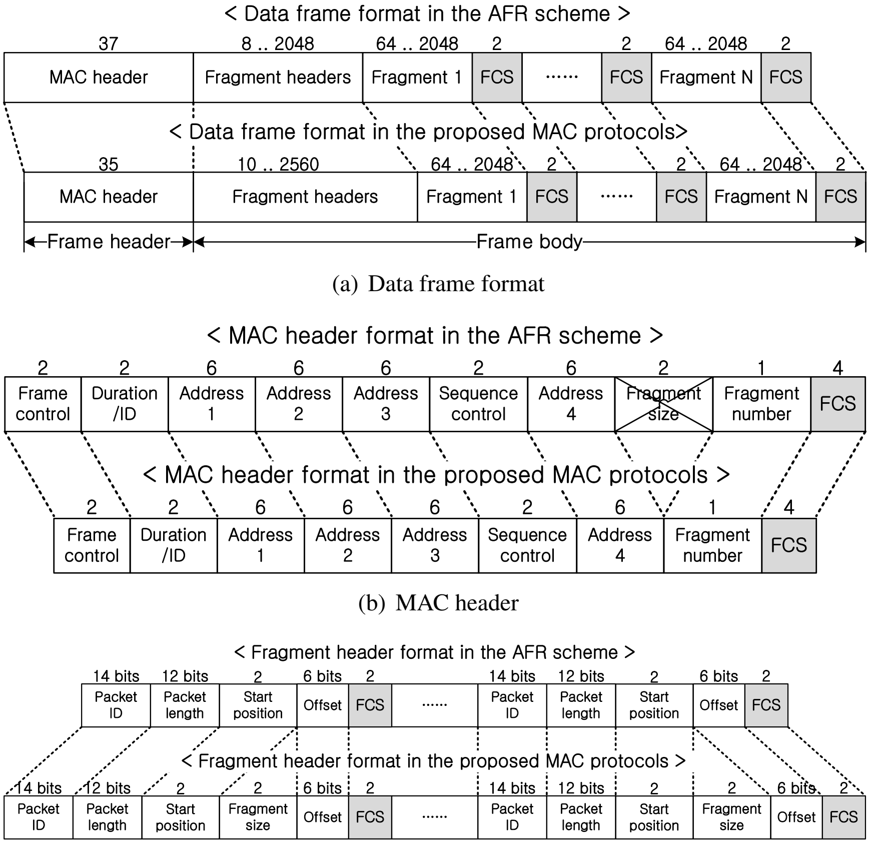
This dissertation is organized as follows. Chapter 2 overviews the aggregation with fragment retransmission (AFR) scheme [4], which is closely related to the proposed MAC protocols, and shows how the transmission duration can be precisely controlled. In Chapter 3, we propose Transmission Order Deducing MAC (TOD-MAC) for maximizing the MAC efficiency in a single cell wireless network. In Chapter 4, Max-min Air-time Fairness MAC (MAF-MAC) is proposed to improve max-min air-time fairness in IEEE 802.11 ad-hoc networks. Finally, we conclude the dissertation in Chapter 5.

Chapter 2

Precisely adjusting transmission duration in IEEE 802.11n WLANs

In this chapter, we briefly introduce the Aggregation with Fragment Retransmission (AFR) scheme, which forms the basic element for the proposed MAC protocols in this dissertation. Then, we explain how transmission duration can be precisely adjusted in the proposed MAC protocols. In [4], the authors proposed a new fragmentation technique, called the AFR scheme. In the AFR scheme, a frame is composed of fragments, and packets that exceed the predetermined fragment size are divided into fragments. A Fragment, rather than a packet, is the basic unit used in retransmission rather (see Fig. 2.1(a)).

In the AFR scheme, packets are divided and combined following the procedure described below [4]. In the MAC header of the AFR scheme, the fields of the MAC header of IEEE 802.11 remain unchanged, except that three additional fields - $fragment\ size$, $fragment\ number$, and $spare$ are added. The $fragment\ size$ and $fragment\ number$ represent the size of a fragment and the number of fragments in a MAC frame, respectively. Each fragment header



(a) Data frame format

(b) MAC header

(c) Fragment header

Figure 2.1 Modified data format in the proposed MAC protocols (The numbers are in bytes except for those explicitly denoted in bits)

is composed of six fields: packet ID (pID), packet length (pLEN), startPos, offset, spare, and FCS. pID and pLEN represent the corresponding ID and length of packet P to which a fragment belongs. The startPos indicates the position of a fragment body in a frame, and the offset is used to record the position of a fragment in packet P. After receiving a frame, the receiver combines the fragments corresponding to packet P based on the information in the MAC and fragment headers (see [4] for the process by which the packet is combined by a receiver) and sends successfully combined packets to the upper layer. The receiver also transmits an ACK frame in the format of Fig. 2.2, where a 32-byte bitmap is simply added to the legacy ACK format. Each

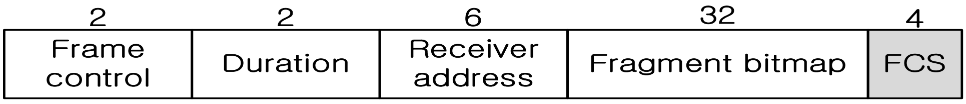


Figure 2.2 ACK frame format

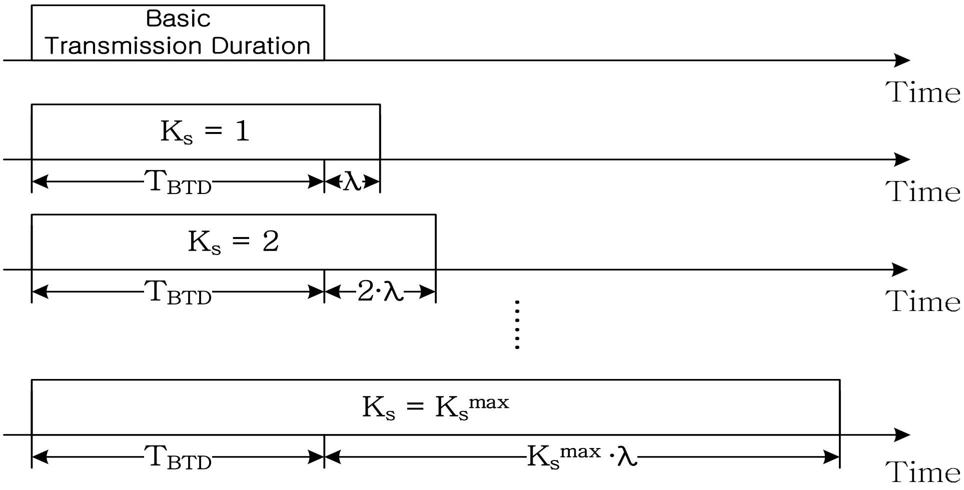


Figure 2.3 The required information coding using transmission duration

bit in the bitmap indicates the correctness of a fragment, from which a sender can retransmit only the corrupted fragments.

If packets can be divided into small fragments, the aggregated frame size can be adjusted at the fragment level, and the transmission duration can also be controlled so that it plays the role usually carried out by control messages. Figure 2.3 illustrates how each node indirectly announces necessary information without incurring any overhead in the proposed MAC protocols. This idea is simple, but very effective to enhance the network performance by exchanging necessary information without overheads. In the proposed MAC protocols, each node announces a positive integer K_s that indicates its present state (transmission order or channel resource usage) and transmits a frame according to K_s. To implicitly announce its K_s, each node calculates its transmission duration as

$$T_{send} \quad = T_{hdr}^{phy} + T_{hdr}^{mac} + T_{hdr}^{frag} + T_p = \quad T_{BTD} + K_s * \lambda, \tag{2.1}$$

where T_{hdr}^{phy}, T_{hdr}^{mac}, T_{hdr}^{frag}, and T_p represent the time duration to transmit the PHY, MAC, fragment headers, and data frame, respectively. And, T_{BTD} is the $basic\ transmission\ duration$ (a fixed parameter), and λ is the difference in duration between $K_s = k$ and $K_s = k + 1$. T_{BTD} in the proposed schemes is set sufficiently shorter than the channel coherence time [1] so that the channel noise level does not change much during a frame transmission. This prevents miss-detection of channel state caused by large fluctuation of channel noise during a frame transmission, and thus we can assume that each node knows the frame transmission time exactly based on physical carrier sensing[2]. After determining the transmission duration, the frame body size S_{fb}, which is the size of the payload and fragment headers, can be calculated depending on the transmission rate R, i.e.,

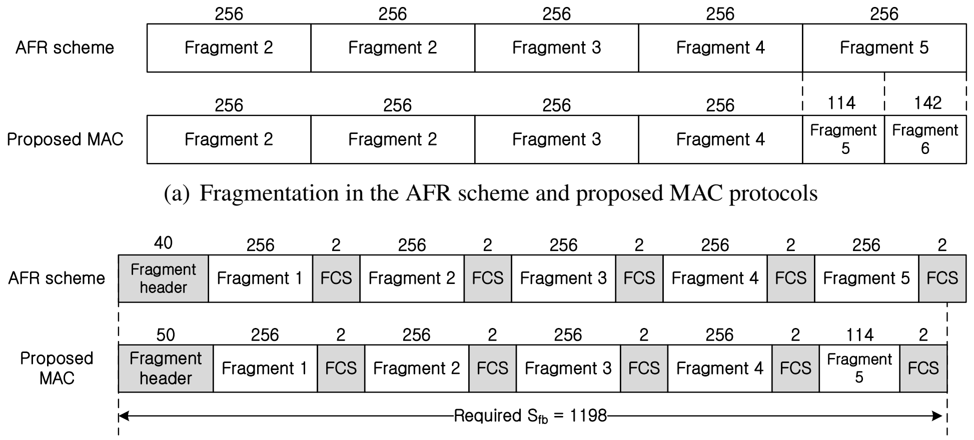
$$S_{fb} \quad = \quad \frac{T_{send} * R}{8} - S_{hdr}^{phy} - S_{hdr}^{mac},$$

where S_{hdr}^{phy} and S_{hdr}^{mac} are the sizes of the PHY and MAC headers, respectively. Each node can subsequently deduce K_s corresponding to the current transmission from T_{send} by (2.1).

Although the fragment aggregation in the AFR scheme is more flexible than the packet aggregation in IEEE 802.11n for controlling transmission duration, it is not sufficient to accurately adjust the transmission duration because, except for the last fragment of a packet, the fragment size is fixed. For example, let T_{BTD} be 192 μs, λ be 4 μs, R be 65 Mbps, packet size S_p be 1280

[1] We consider that the coherence time of a typical WLAN channel is on the order of a few tens of milliseconds [10].

[2] The physical carrier sensing mechanism defined by IEEE is known as clear channel assessment (CCA). The CCA determines the channel state (busy or idle) in PHY layer, and it takes 4μs to judge the channel state in 802.11n [11, 12].



(a) Fragmentation in the AFR scheme and proposed MAC protocols

(b) Fragment aggregation inthe AFR scheme and proposed MAC protocols

Figure 2.4 Fragment techniques in the AFR scheme and proposed MAC protocols

bytes (B) and fragment size S_{frag} be 256 B, which shows near-optimal performance across a wide range of BERs in the AFR scheme. When a packet of 1280 B is divided into five fragments (256; 256; 256; 256; 256), a node with $K_s = 1$ must transmit for 196 μs to indirectly announce that its K_s is 1. According to IEEE 802.11n, the time duration to transmit a PHY header is 44 μs. Therefore, the time duration to transmit the payload and MAC header (37 B in the AFR scheme) needs to be $192 + 4 - 44 = 152$ μs when we use a single antenna radio. That is, S_{fb} must be 1198 B. Since 1198 is not a multiples of $256 + 8 + 2 = 266$ (8 B for the fragment header and 2 B for the FCS of a fragment in the AFR scheme), a sender cannot make a frame that meets the required T_{send} using the AFR scheme. The S_{fb} size should be either 1064B or 1330B, and thus it cannot precisely adjust the transmission duration to inform its K_s in the AFR scheme. To adjust the transmission duration more precisely, the size of each fragment needs to vary, which requires some modification of the AFR scheme.

Figure 2.1 shows the difference in the data frame format between the AFR scheme and proposed MAC protocols (the ACK frame format is the same for both the proposed MAC protocols and the AFR scheme). In the proposed MAC protocols, the $Fragment\ size$ field is in the fragment header instead of in the MAC header. In this way, each fragment can vary in size, and consequently the transmission duration can be precisely controlled in the proposed MAC protocols. For the required S_{fb} of 1198 B in the previous example, a packet can be divided into six fragments (256; 256; 256; 256; 114; 142;) and then the node can make a frame of length $S_{fb} = (256 + 10) * 4 + (114 + 10) + (2 * 5) = 1198$ B (10 B for the fragment header and 2 B for the FCS of a fragment). Figure 2.4 shows the difference in the fragmentation technique used between the AFR scheme and proposed MAC protocols.

Algorithm 1 Pseudo Code of receiver's running logic in the proposed MAC protocols

```
if MAC header is correct then
    for $i = 0$ to $fragment\ number$ - 1 do
        if Fragment $i$'s header is correct then
            if $pLEN < fgLEN$ then
                fragment $i$'s length = $pLEN$
            else
                fragment $i$'s length = $fgLEN$
            end if
            fragment start position = $startPOS$ in the fragment header.
            check the correctness of the fragment body using its $FCS$.
        end if
        record the correctness (including the fragment header and fragment body) of each fragment in an ACK bitmap.
    end for
    construct an ACK frame using the ACK bitmap and transmit it back to the sneder.
    update the receiving queue according to the ACK bitmap.
    check the receiving queue and transfer all correctly received packets upwards, and remove them from the receiving queue.
else
    discard this frame and defer an EIFS before initiating the next transmission.
end if
```

After receiving a frame, the receiver operates as shown in Algorithm 1, which is similar to the receiver algorithm of the AFR scheme, to reconstruct packets from the fragments. The zero-padding is used when the difference between the required S_{fb} and a packet is smaller than the fragment header size S_1 (= 10 B), i.e., $(S_{fb} - S_p) < S_1$. Using the modified fragmentation technique and zero padding, the frame size can be precisely adjusted so that each transmission lasts for specified duration.

Chapter 3

Improving the MAC efficiency of IEEE 802.11n WLANs

3.1 Background and Related Work

As the demand for multimedia applications such as HDTV (20 Mbps) and DVD (9.8 Mbps) rapidly increases in wireless LANs (WLANs), higher bandwidth is required in wireless networks. Recent IEEE 802.11n proposals seek to support physical (PHY) layer rates of up to 600 Mbps [1], and IEEE 802.11ac/ad aims to support PHY layer rates in the Gbps range [2, 3]. Although the transmission rate may significantly increase, one does not see a commensurate increase in user throughput because the MAC efficiency of IEEE 802.11 rapidly decreases with increasing PHY rates [4, 5]. This is because even though increasing PHY rates leads to faster transmission of the MAC frame payloads, overheads such as PHY headers and contention time typically do not decrease at the same rate. In transmitting a frame, the proportion attributed to overhead becomes larger as the PHY rate increases. According to a study conducted by *Li et al.* [4], MAC efficiency falls from 42% at 54 Mbps PHY rate to only 10% at 432 Mbps PHY rate.

Many schemes have been proposed to improve MAC efficiency in very high-speed WLANs. In [13] and [14], the backoff process is modified to reduce the number of collisions and idle slots, which are the main causes of overhead during channel contention. However, the MAC efficiency in high-speed networks is intolerably low even in ideal cases, where there are no channel errors or collisions [4, 5]. In the Block ACK [5, 15] and Burst ACK [16–18] schemes, several packets can be transmitted at the end of a backoff process, and so the number of ACKs and SIFSs can be reduced. A backoff process is performed to transmit a series of data and ACKs in Burst ACK, whereas Block ACK goes one step further by using a single ACK frame for multiple data frames. In this way, the contention time and ACK time per packet can be reduced and, consequently, the MAC efficiency can be improved. However, the PHY and MAC header overheads are left untouched in these schemes, and the time to transmit a frame quickly becomes dominated by PHY headers as the PHY rate increases.

IEEE 802.11n [1] introduced several mechanisms including frame aggregation and Block ACK to enhance MAC efficiency. The standard supports two types of frame aggregation schemes, the aggregated MAC level service data unit (A-MSDU) and the aggregated MAC layer protocol data unit (A-MPDU). Several analytic studies have investigated the effect of A-MPDU and/or A-MSDU frame sizes on the achievable throughput and their optimal sizes to improve throughput [19–22]. In addition, a hybrid frame aggregation scheme known as two level aggregation, which combines A-MSDU and A-MPDU, has been introduced [23, 24]. Even for such features in IEEE 802.11n, MAC efficiency can become severely degraded in noisy channels as packet sizes increase.

To overcome this shortcoming, Li *et al.* proposed the Aggregation with Fragment Retransmission scheme (AFR scheme) [4], which modifies the frame aggregation algorithm in IEEE 802.11n. In the AFR scheme, packets that exceed a fragment size are divided into fragments to form a frame, and fragments, rather than an entire frame or packets, are retransmitted. Thus, the AFR scheme gives good throughput performance even in noisy channels regardless of packet size. Furthermore, transmission delays are minimized by using a zero-waiting mechanism in which frames are transmitted immediately once a node wins a transmission opportunity. However, the MAC efficiency of the AFR scheme is lower than that of IEEE 802.11n in low noisy channels because of unnecessary frame aggregation. Furthermore, the AFR scheme does not reduce the time for contention including idle time and collisions. This is one of the main reasons for low efficiency, and as a consequence, the efficiency decreases as the number of nodes increases.

In this paper, we propose the Transmission Order Deducing MAC (TOD-MAC) protocol for CSMA/CA wireless networks, which is a novel approach that improves MAC efficiency in very high-speed WLANs by adopting frame aggregation and minimizing contention time. We assume that all the nodes are located in a carrier sensing range of each other. In the traditional MAC protocol, the length of a packet or a frame is usually fixed and the transmission duration is determined by the data rate. However, we show how to determine the transmission order of the nodes in a WLAN using transmission duration. If packets can be divided into small fragments, the aggregated frame size can be adjusted at the fragment level, and the transmission duration can also be controlled so that it plays the role usually carried out by control messages. This idea is simple, but very effective as each node can indirectly announce its transmission order without incurring any overhead. In TOD-MAC, a node transmits a frame in a round robin manner based on the transmission order, which can minimize the idle time between two consecutive transmissions without causing a collision. To correctly announce its transmission order, a node in TOD-MAC adjusts the size of each fragment, unlike in the AFR scheme. In this way, TOD-MAC achieves a much higher MAC efficiency than the AFR scheme or IEEE 802.11n, regardless of the number of nodes. Furthermore, high MAC efficiency is maintained even in noisy channels because TOD-MAC uses fragments as the basic unit of retransmission.

Recently, several MAC protocols was proposed to implement the round robin access method [25–27]. However, these schemes required either additional control messages or sub-headers for proper operation of the round robin access method. These induce additional overhead that severely degrades MAC efficiency because these are transmitted at the basic rate as the PHY header, and all the nodes need to be in the transmission range of each other, not the carrier sensing range as in TOD-MAC, to obtain necessary information from control messages or sub-headers. Furthermore, they did not consider the frame aggregation, which could be one of the key features for improving MAC efficiency.

The remainder of this chapter is organized as follows. Section 3.2 overviews the AFR scheme, which is closely related to TOD-MAC. Section 3.3 explains the operation of TOD-MAC that improves MAC efficiency. Section 3.4 evaluates TOD-MAC and compares it to other MAC schemes for very high-speed WLANs [1,4]. Section 3.5 concludes this chapter.

3.2 Preliminaries

In this section, we briefly introduce the Aggregation with Fragment Retransmission (AFR) scheme, which forms the basis for TOD-MAC. In [4], the authors presented two basic requirements that must be met by any aggregation schemes: zero-waiting for aggregation and high MAC efficiency. Zero-waiting for aggregation means that frames are immediately transmitted whenever a node wins a transmission opportunity, without it waiting for packets to be accumulated at the MAC layer.

We discuss MAC efficiency and how to achieve high MAC efficiency in the AFR scheme in more detail as it is closely related to the development of TOD-MAC. In [4], the MAC efficiency for packet P is defined as

$$\eta_p = \frac{T_p}{T_p + T_{oh}^p} = \frac{S_p/R}{S_p/R + T_{oh}^p} \tag{3.1}$$

where T_p and T_{oh}^p are the times required to transmit a packet (i.e., a payload) and overheads, respectively. As PHY rate R increases, for a fixed packet size S_p, the time T_p $(= S_p/R)$ to transmit a packet decreases. If T_{oh}^p does not decrease, then the efficiency η_p decreases to zero as R increases to infinity.

From (3.1), one can see that it is indeed possible to maintain a high MAC efficiency as R increases, if T_{oh}^p decreases in inverse proportion to R. Considering T_{oh}^p in more detail, it can be decomposed into the following elements:

$$T_{oh}^p = \frac{(T_{hdr}^{phy} + T_{hdr}^{mac} + T_{hdr}^{frag} + T_{CW} + T_{ack}) \cdot r}{M}, \tag{3.2}$$

where T_{hdr}^{phy}, T_{hdr}^{mac}, T_{hdr}^{frag}, and T_{ack} represent the time duration to transmit the PHY, MAC, fragment headers, and ACK frame, respectively. T_{CW} is the contention time to win an opportunity for transmitting a frame, and r denotes the average number of transmissions required to successfully transmit all the fragments of a packet. For a given fragment size S_{frag}, the number of fragments in a frame n_{frag} increases with the number of packets M in the frame, i.e., $n_{frag} = n'_{frag}M$, where n'_{frag} is the number of fragments per packet. Assuming that the number of packets M in a frame is proportional to R (i.e., $M = bR$), then we have $n_{frag} = n'_{frag}bR$. Therefore, $T_{hdr}^{frag}/M = n'_{frag}S_1/R$, where S_1 is the fragment header size, and consequently, the MAC efficiency per packet is given by

$$\eta_p = \frac{S_p}{S_p + r \cdot (a/b + n'_{frag} \cdot S_1)}, \tag{3.3}$$

where $a = T_{hdr}^{phy} + T_{hdr}^{mac} + T_{CW} + T_{ack}$. If factor b increases, η_p asymptotically approaches

$$\tilde{\eta_p} \;=\; \frac{S_p}{S_p + r \cdot m' \cdot S_1} = \frac{1}{1+d} \tag{3.4}$$

where $d = (rn'_{frag}S_1)/S_p$.

This shows that the efficiency is fundamentally limited by the number of fragments per packet n'_{frag} and the number of retransmissions r. If we use a large fragment size, which corresponds to a small n'_{frag}, then the fragment is more likely to become corrupted in a noisy channel. On the other hand, if a packet is divided into many smaller fragments, the probability of a fragment being corrupted is low, and thus n'_{frag} becomes larger and r gets smaller. If the fragment size S_{frag} can be adjusted according to channel conditions, high MAC efficiency can be achieved. In [4], the authors proposed a new fragmentation technique, which is called the AFR scheme and explained in Chapter 2, for improving MAC efficiency. In the AFR scheme, a frame is composed of fragments, and packets that exceed the fragment size are divided into fragments. Fragments are the basic unit used in retransmission rather than frames (see Fig. 2.1(a)).

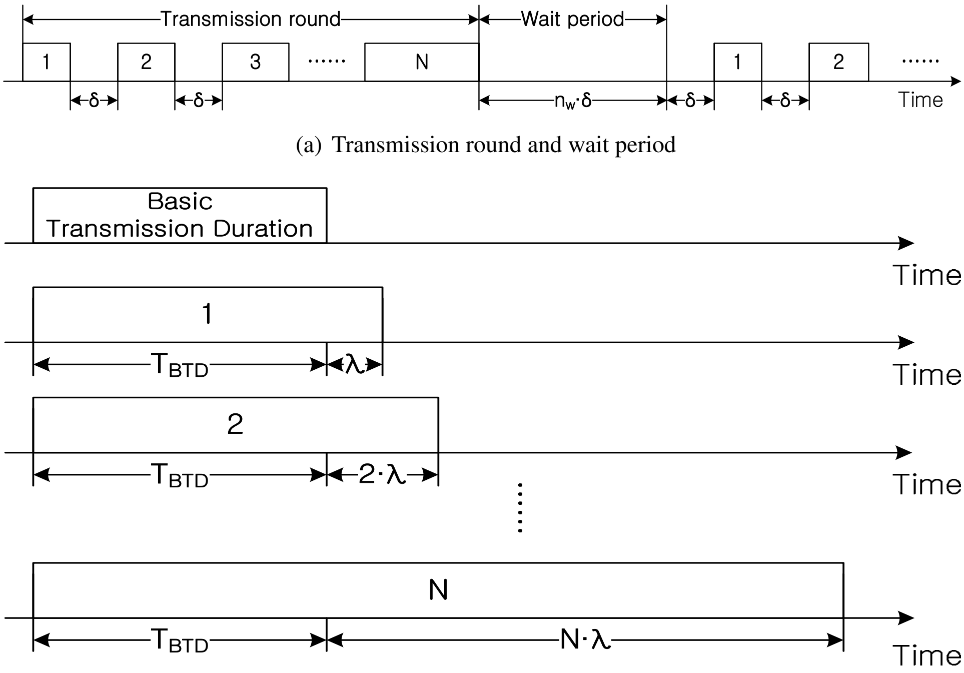
In [4], the authors assume that a $(= T_{hdr}^{phy} + T_{hdr}^{mac} + T_{CW} + T_{ack})$ can be ignored when b is large enough. In this case, the MAC efficiency can be improved by controlling the values of n'_{frag} and r. However, the value of b can also be small. Furthermore, the number of retransmissions r can increase depending on channel conditions and the collision rate when there are many transmitters in a network. Therefore, $r \cdot a/b$ in (3.3) must not be ignored in practice, and so there is an upper bound on how much the MAC efficiency can be improved by the AFR scheme. TOD-MAC improves the MAC efficiency by minimizing a, especially T_{CW}, and r. In the following section, we explain the operation of TOD-MAC that achieves higher MAC efficiency.

3.3 Transmission Order Deducing (TOD) MAC protocol

As the PHY rate increases, the time to transmit a frame is quickly dominated by a fixed overhead associated with the PHY header, contention time, etc. Minimizing the wasted time caused by collisions or channel errors is crucial to improve the MAC efficiency. Data transmission in a round robin manner, instead of contention for an opportunity for data transmission, is an attractive alternative. If each node transmits in a round robin manner, the contention time and collision rate can both be minimized at the same time. That is, the values of a and r in (3.3) decrease and, consequently, the MAC efficiency can be improved. However, a round robin method is not compatible with the distributed nature of IEEE 802.11 DCF, in which the transmission order of nodes is determined by contention. It usually requires control messages to inform the nodes in a network of the transmission order. Moreover, when a node newly joins or leaves the network, each node needs to know its transmission order, which can change over time. Thus, it is hard to operate appropriately in a dynamic environment without control messages.

The Transmission Order Deducing MAC (TOD-MAC) protocol is based on a round robin scheme, but it does not use any control messages. The main idea underlying TOD-MAC is that each node that is located in carrier sensing range of other nodes implicitly announces its transmission order by way of its transmission duration. Each node determines the transmission order of the other nodes by measuring the duration of their transmissions. Using a fragment aggregation technique similar to that of the AFR scheme, which divides packets into fragments of a small fixed length, the transmission duration is no longer solely dependent on the transmission rate and can be flexibly adjusted in TOD-MAC.

Figure 3.1 illustrates the basic concept underlying TOD-MAC when there are N nodes in a network. In the figure, δ is slot time and n_w is a design parameter. In this paper, the time period during which each node in a network transmits a frame is called a $transmission\ round$ and the



(a) Transmission round and wait period

(b) Transmission order coding

Figure 3.1 Basic concept underlying TOD-MAC

time period between two consecutive transmission rounds is called the $wait\ period$. Each node has a transmission order O (marked in each square box in Fig. 3.1), which changes from round to round, and transmits a frame according to its transmission order. To implicitly announce its order, each node calculates its transmission duration as

$$T_{send} \;=\; T_{BTD} + O * \lambda, \qquad (3.5)$$

where $T_{send} = T_{hdr}^{phy} + T_{hdr}^{mac} + T_{hdr}^{frag} + T_p$ is the time duration to send a data frame composed of fragments, T_{BTD} is the $basic\ transmission\ duration$ (a fixed parameter), and λ is the difference in duration between the kth and $(k+1)$th transmissions. Figure 3.1(b) shows how each node determines its T_{send} according to its order, O. After determining the transmission duration, the frame body size S_{fb}, which is the size of the payload and fragment headers, can be calculated in terms of the transmission rate i.e.,

$$S_{fb} \;=\; \frac{T_{send} * R}{8} - S_{hdr}^{phy} - S_{hdr}^{mac}, \qquad (3.6)$$

where S_{hdr}^{phy} and S_{hdr}^{mac} are the sizes of the PHY and MAC headers, respectively. Each node can subsequently deduce the transmission order corresponding to the current transmission from T_{send} when all the nodes are in carrier sensing range of each other.

In order for TOD-MAC to use the transmission order information to achieve high throughput performance and air-time fairness without collisions, we have to resolve the following questions:

- How can transmission duration T_{send} be accurately adjusted according to the transmission order?
- How can the transmission order be deduced in a distributed manner?
- How can situations where a node newly joins or leaves a network be accommodated?

Table 3.1 Parameters of node i for transmission order determination.

CW_i	Contention window of node i
BC_i	Backoff counter of node i
WT_i	The number of idle slots to wait for node i before starting to decrease its BC
O_i	Transmission order of node i
$O_{min,i}$	Minimum transmission order value among the transmission orders announced to node i
$O_{max,i}$	Maximum transmission order value among the transmission orders announced to node i
m_i	Number of transmissions between the transmissions of $n(min)$ node and node i

- How are the cases of transmission collisions and unsaturated nodes to be handled?

3.3.1 Adjusting the transmission duration

In high-speed WLANs such as IEEE 802.11n/ad/ac, packet aggregation is adopted to reduce transmission overheads. Although fragment aggregation in the AFR scheme is more flexible than packet aggregation in IEEE 802.11n for controlling transmission duration, it is not sufficient to accurately adjust the transmission duration because, except for the last fragment of a packet, the fragment size is fixed. In TOD-MAC, the $Fragment\ size$ field is in the fragment header instead of in the MAC header as previously stated in Chapter 2. In this way, each fragment can have a different size and, consequently, the transmission duration can be precisely controlled in TOD-MAC. Furthermore, zero-padding is used when the difference between the required S_{fb} and a packet is smaller than the fragment header size S_1 (= 10 B), i.e., $(S_{fb} - S_p) < S_1$. Using the modified fragmentation technique and zero padding, frame size can be precisely adjusted and each transmission can last according to the specified duration. After receiving a frame, the receiver can easily reconstruct packets from the fragments.

3.3.2 Arranging transmission order

Let us now look at how the transmission order is determined distributively in TOD-MAC so that the transmission order is arranged as in Fig. 3.1(a). In a network consisting of N nodes, node i

Table 3.2 Updating the parameters of node i for each idle slot.

Event Parameters	Normal node			Collision / Unsaturated node		
	$WP_i > 0$	$WP_i = 0$		$WP_i > 0,$ $I \geq 3$	$WP_i > 0,$ $I < 3$	Otherwise
		$O_i = 0$	$O_i > 0$			
O_i	-	$m_i + 1$	-	-	-	-
BC_i	-	O_i	$BC_i - 1$	$BC_i - 1$	-	-
$O_{min,i}$	-	O_i	-	-	-	-
$O_{max,i}$	-	O_i	-	-	-	-
WP_i	$WP_i - 1$	-	-	$WP_i - 1$	$WP_i - 1$	-

needs CW_i, BC_i, O_i, $O_{min,i}$, $O_{max,i}$, WP_i, and m_i, which are listed in Table 3.1, in order to calculate the transmission order. CW_i, BC_i, and O_i are the contention window, backoff counter, and transmission order of node i, respectively. O_{min} and O_{max} are the minimum and maximum transmission orders in the network, respectively, and $O_{min,i}$ and $O_{max,i}$ are the minimum and maximum transmission orders deduced by node i. WP_i is the number of idle slots that node i waits before starting to decrease BC_i, and node i deduces that the wait period ends in WP_i slots. WP_i is decremented by one for each idle slot, and is reset to $n_w \cdot \delta$ when $O_{max,i}$ is updated or the transmission of node i ends (this will be explained in detail later). After WP_i becomes zero, BC_i is decreased by one for each idle slot as shown in Table 3.2. Let us call node i whose transmission order O_i is equal to k ($O_i = k$) as $n(k)$, and denote the nodes that have minimum transmission order (O_{min}) and maximum transmission order (O_{max}) as $n(min)$ and $n(max)$, respectively. For example, assume that there are only two nodes in a network, nodes i and j with $O_i = 3$ and $O_j = 5$. Then, node i is node $n(min)$ and node j is node $n(max)$, and O_{min} and O_{max} are three and five, respectively. We also denote m_i as the number of transmissions between the transmissions of node $n(min)$ and node i.

Table 3.3 Updating the parameters of node i at the end of transmission by $n(k)$.

<table>
<tr><th rowspan="2">Event
Parameters</th><th colspan="3">node $i \neq n(k)$</th><th colspan="3">node $i = n(k)$</th></tr>
<tr><th>$O_{max,i} < k$</th><th>$k < O_{min,i}$</th><th>Otherwise</th><th colspan="2">TX success</th><th>TX failure</th></tr>
<tr><td>CW_i</td><td>-</td><td>-</td><td>-</td><td colspan="2">-</td><td>$2 * CW_i$</td></tr>
<tr><td rowspan="2">O_i</td><td rowspan="2">-</td><td rowspan="2">-</td><td rowspan="2">-</td><td colspan="2">Change by (3.7)</td><td rowspan="2">Select from
$[1, CW_i]$</td></tr>
<tr><td>$O_i > 0$</td><td>$O_i = 0$</td></tr>
<tr><td>BC_i</td><td>O_i</td><td>-</td><td>-</td><td>O_i</td><td>$O_{max,i}$</td><td>O_i</td></tr>
<tr><td>$O_{min,i}$</td><td>-</td><td>k</td><td>-</td><td>O_i</td><td>1</td><td>128</td></tr>
<tr><td>$O_{max,i}$</td><td>k</td><td>-</td><td>-</td><td>O_i</td><td>1</td><td>0</td></tr>
<tr><td>WP_i</td><td>$n_w \cdot \delta$</td><td>-</td><td>-</td><td>$n_w \cdot \delta$</td><td>$n_w \cdot \delta$</td><td>-</td></tr>
<tr><td>m_i</td><td>$m_i + 1$</td><td>0</td><td>$m_i + 1$</td><td>0</td><td>0</td><td>0</td></tr>
</table>

When node i newly joins the network, it selects its transmission order O_i from $[1, CW_i]$, sets BC_i, $O_{min,i}$, and $O_{max,i}$ equal to O_i, WP_i to $n_w \cdot \delta$, and m_i to zero. Each node needs to know the following information to transmit successfully in a round robin manner: 1) the time when the *current* transmission round ends, 2) the time when the *next* transmission round starts, and 3) its proper transmission order in the next round. (In a network of N nodes, we say that the transmission order is proper when the transmission orders of nodes are in $[1, N]$ without overlap.) Using this information, we try to make $WP_i = n_w \cdot \delta$ and BC_i to O_i for all i at the end of the transmission round so that each node can transmit in proper order at the next transmission round. However, it is not easy to obtain this information because it can change whenever a node newly joins or leaves the network.

Let us look at how each node can deduce the time when the *current* round ends and also the time when the *next* round starts. Basically, nodes $n(min)$ and $n(max)$ transmit first and last in a transmission round, respectively. Thus, node i, $i = 1, 2, \ldots, N$, can deduce the start (and the end) of a round by comparing $O_{min,i}$ ($O_{max,i}$) with the order deduced from the duration of the current transmission following the procedures described below. When there is a transmission by $n(k)$ and k is larger than $O_{max,i}$, node i updates $O_{max,i}$ to k, BC_i to O_i, and WP_i to $n_w \cdot \delta$, and increases m_i by one. After updating $O_{max,i}$, it starts to decrease WP_i by one for each idle slot. If WP_i becomes zero and there is no more updating of $O_{max,i}$, node i concludes that the $current$ round has ended. In a similar way, whenever node i discovers, after the transmission by $n(k)$, that k is smaller than the current $O_{min,i}$, it updates $O_{min,i}$ to k and assumes that the $next$ round has started with the transmission of $n(k)$ and sets m_i to zero. If $O_{min,i} \leq k \leq O_{max,i}$, node i simply increases m_i by one without updating $O_{min,i}$ or $O_{max,i}$. The way in which the node parameters are updated in accordance with various events is summarized in Table 3.3.

When node i successfully transmits a data frame, it calculates its transmission order O_i using

$$O_i \quad \leftarrow \quad \begin{cases} O_i - 1, \text{ if } O_i = O_{min,i} \text{ and } O_i > 0, \\ \\ O_{min,i} + m_i, \text{ if } O_i \neq O_{min,i} \text{ and } O_i > 0. \end{cases} \tag{3.7}$$

As stated previously, when node i newly joins a network, it randomly selects its transmission order O_i from $[1, CW_i]$. Thus, there is a possibility that O_{min} will be greater than one in a transient state. If O_{min} is l (> 1), there must be a node whose transmission order is greater than or equal to $N + l$, which needs to be changed to a number between one and N so that the transmission orders are properly arranged. Therefore, if O_i is equal to $O_{min,i}$ and the transmission of node i is successful, it deduces that it has become $n(min)$ in this transmission round, and therefore decreases O_i by one as long as it is greater than zero. Otherwise, it sets O_i to $O_{min,i} + m_i$. In this way, O_i for each node can be made between one and N without overlap when the transmission order of $n(min)$ is one. After node i determines O_i using (3.7), it resets

$O_{min,i}$ and $O_{max,i}$ to O_i so that node i can follow the changes of O_{min} and O_{max} in a dynamic environment. In addition, node i sets BC_i to O_i and WP_i to $n_w \cdot \delta$.

Note that $n(k)$ increases its transmission duration as k increases because it must transmit for the duration of $T_{BTD} + k \cdot \lambda$. This means that $n(k + 1)$ transmits longer than $n(k)$ by λ and, consequently, the air-time fairness can become severely degraded if the transmission order does not change. To achieve air-time fairness, each node adjusts its transmission order after each transmission round. If the transmission order of each node is adjusted using (3.7), it is decremented by one after each transmission round. Note that the transmission order of node $n(min)$ naturally becomes zero first. Node l, if it is currently node $n(min)$, must find its proper transmission order (O_l^p) for the next round when O_l becomes zero. Thus, node l sets BC_l to $O_{max,l}$, not to O_l ($= 0$ in this case), and sets m_l to zero. After this, it temporarily sets both $O_{max,l}$ and $O_{min,l}$ to one. These will be updated when WP_l becomes zero, i.e., when the wait period ends. Table 3.3 summarizes the way in which each node updates its parameters at the time of successful transmission.

Since a transmission collision occurs if O_l^p is less than or equal to O_{max} in the next round, the value of O_l^p must be larger than O_{max} in the next round. However, it is difficult to know the value of O_{max} in the next round because it can change whenever a node newly joins or leaves the network. In order to determine the value of O_l^p, node l counts the number of transmissions in the current transmission round. Since $O_{min,l}$ is set to one, node l does not update $O_{min,l}$. The value of m_l is increased by one whenever there is a transmission before WP_l becomes zero. In this case, the value of $m_l + 1$ at the end of the wait period after the current transmission round becomes the maximum possible value for O_{max} in the next round. For example, assume that there are N nodes in the current round, and N_1 nodes join the network, with each transmitting a frame during the wait period. If any node leaves the network in the next round, the value of O_{max} in the next round will be less than $N + N_1$, which is equal to $m_l + 1$. Otherwise, O_{max}

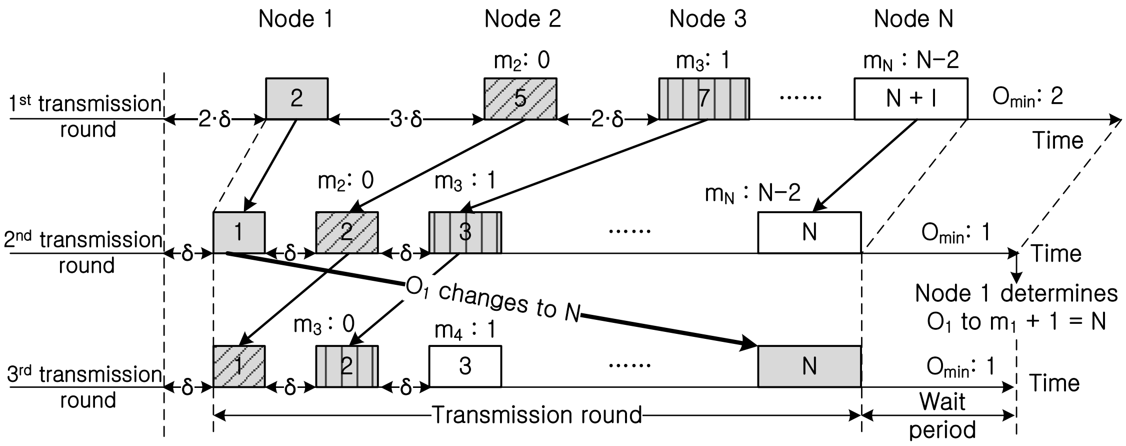


Figure 3.2 Determining transmission order

will be equal to $N + N_1$. After determining the transmission order O_l, node l sets BC_l, $O_{min,l}$, and $O_{max,l}$ to newly determined O_l. Table 3.2 summarizes the way in which node i updates its parameters according to the value of O_i when WP_i becomes zero.

Let us now look at some examples for the following cases: 1) How TOD-MAC operates at the time of network initialization, 2) how each node determines when the current round ends and when the next round starts, 3) how each node transmits in round robin manner, and 4) how TOD-MAC operates when a node newly joins or leaves the network.

How TOD-MAC operates at the time of network initialization

Figure 3.2 shows how nodes determine their transmission orders when they form a network. In the first round, node i, $i = 1, 2, \ldots, N$, randomly selects its transmission order O_i from the range$[1, CW_i]$ (i.e., $2, 5, 7, \ldots, N + l$ in Fig. 3.2), and sets BC_i to O_i. Here, we do not take collisions into consideration, i.e., we assume that each node initially selects a value for its transmission order that is different from that of the others. (The case of transmission collision will be discussed later.) Presently, node 1, which is $n(2)$, is also $n(min)$ in the network, and thus it transmits first in the first round. After the transmission by node 1, it notices that $O_1 =$

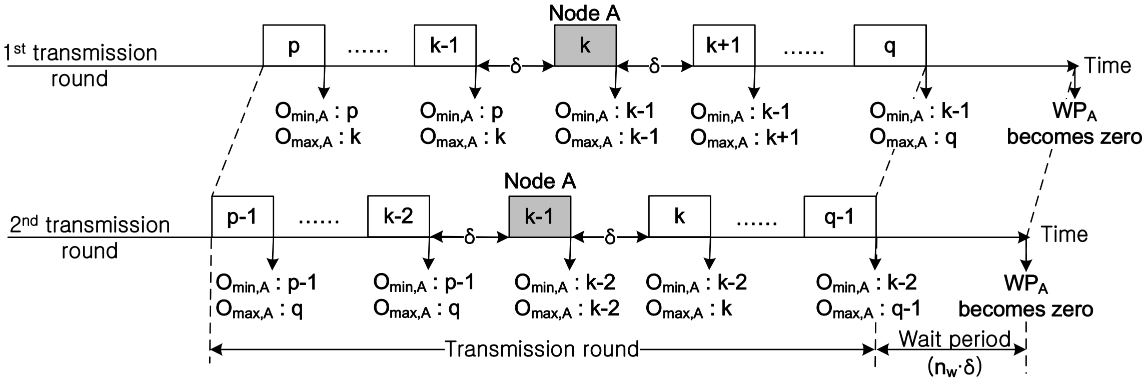


Figure 3.3 Determination by node A of when the *current* round ends and when the *next* round starts using $O_{min,A}$ and $O_{max,A}$.

$O_{min,1} = 2$, and thus decreases the value of O_1 by one in accordance with (3.7). At this point, node j, $j = 2, 3, \ldots, N$, sets $O_{min,j}$ to two, which is the minimum among the transmission orders calculated from the transmission durations. After successfully transmitting a frame, node j sets its transmission order to $O_{min,j} + m_j = 2 + m_j$. In this way, each node i can accurately determine its proper transmission order in the second transmission round. Node 1 prepares to change its order after transmitting in the second round, because O_1 is now zero. Since there are $N - 1$ transmissions when the second wait period ends, node 1 changes O_1 to $m_1 + 1 = N - 1 + 1 = N$ in the third round. In this way, node i decreases its transmission order by one in accordance with (3.7), if O_i is greater than zero. If O_i is equal to zero, it changes O_i to $m_i + 1$ in the next round and, consequently, air-time fairness is achieved as the rounds proceed.

How each node determines when the current round ends and when the next round starts

Figure 3.3 illustrates how node A (node $n(k)$ in this example) determines when the *current* round ends and when the *next* round starts using $O_{min,A}$ and $O_{max,A}$. Let O_{min} and O_{max} be p and q ($> p$) in the first round, respectively, and assume that the transmission orders of nodes are well arranged from p to q without overlap. Note that node A initially sets $O_{min,A}$

and $O_{max,A}$ equal to its transmission order k. Node A then updates $O_{min,A}$ to p when the transmission by node $n(p)$ ($p < O_{min,A} = k$) ends, and it can easily deduce that the first round has started. After node A transmits a frame, it sets its transmission order O_A to $k-1$, and resets both $O_{min,A}$ and $O_{max,A}$ to $k-1$ in accordance with Table 3.3. When transmission by $n(q)$ ends, node A updates $O_{max,A}$ to q and concludes that the first round has ended when there is no more updating of $O_{max,A}$ and there is no transmission during the wait period. As previously stated, the transmission orders of each node is decremented by one after the first transmission round according to (3.7). Since $O_{min,A}$ has been reset to $k-1$ in the first round, node A updates $O_{min,A}$ to $(p-1)$when the transmission by node $n(p-1)$ ends, and thus it can determine that the second round has started. After its transmission in the second round, it resets both $O_{min,A}$ and $O_{max,A}$ to the newly determined transmission order O_A (=$k-2$). In the second round, node A lastly updates $O_{max,A}$ when the transmission of node $n(q-1)$ ends, and thus it concludes that the second round has ended at the end of the transmission by node $n(q-1)$. In this way, node i can correctly deduce when the $current$ round ends and when the $next$ round starts using $O_{min,i}$ and $O_{max,i}$.

How each node transmits in a round robin manner

In TOD-MAC, each node knows when to transmit in a round robin manner, as shown in Fig. 3.1(a), based on its transmission order and the time when the current round ends. When node i updates $O_{max,i}$ or successfully transmits a data frame, it deduces that the current round can be ended and sets BC_i to O_i and WP_i to $n_w \cdot \delta$. Note that the idle time between two consecutive transmissions is set to a single slot time to maximize MAC efficiency, and the backoff time is determined according to the transmission order; i.e., node i, if it is $n(k)$, sets its backoff counter to $BC_i = O_i = k$ after its successful transmission. If each node i, $i = 1, 2, \ldots, N$, knows when a transmission round has ended and sets BC_i to O_i and WP_i to $n_w \cdot \delta$, then it can transmit

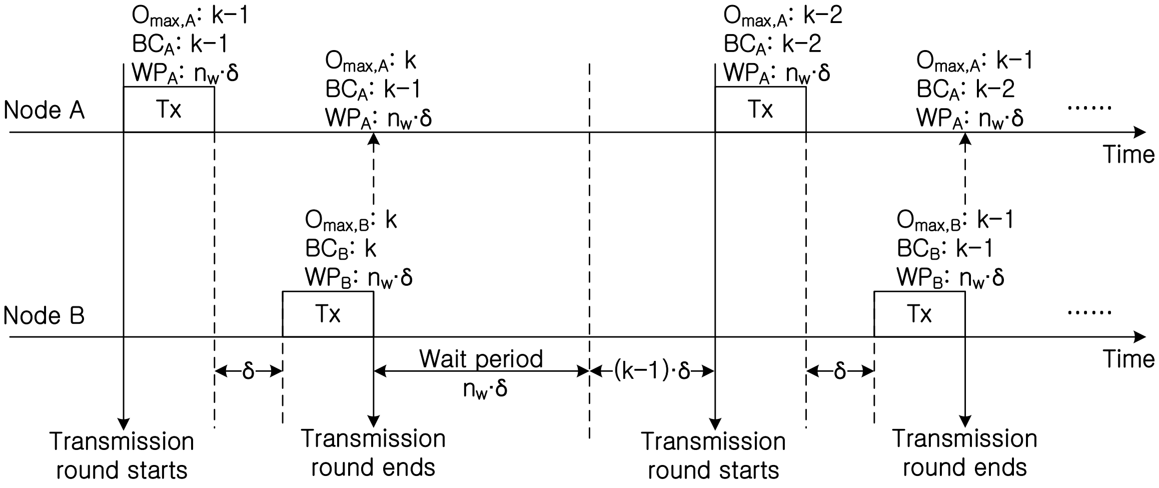


Figure 3.4 Node transmission in a round robin manner using O_{max}.

in a round robin manner without any collision.

For example, let us assume that there are two nodes, A ($O_A = k$) and B ($O_B = k + 1$), in a network and nodes A and B set their backoff counter to k and $k + 1$, respectively. In this case, node A, which is $n(min)$, will transmit before node B. Node A assigns $k - 1$ to O_A in accordance with (3.7), and sets BC_A to $k-1$ and WP_A to $n_w \cdot \delta$ after transmitting a data frame. When node B ends transmission, which is $n(max)$, node A updates $O_{max,A}$ to $(k+1)$, and sets WP_A to $n_w \cdot \delta$ and BC_A to $k-1$. Note that node B also sets O_B to $O_{min,B} + m_B = k$, BC_B to k, and WP_B to $n_w \cdot \delta$ after transmitting a frame. Nodes A and B then both wait the same time period ($WP_A = WP_B$) before starting the backoff process, and the backoff counters of nodes A and B are $k - 1$ and k, respectively, when the transmission of node B ends, which is also the end of the current round. Thus, nodes A and B have to wait $(k + n_w - 1) \cdot \delta$ and $(k + n_w) \cdot \delta$ to transmit data, respectively. That is, node A transmits before node B, and the idle time between transmissions by nodes A and B is a single slot time. Furthermore, since node A always resets $O_{max,A}$ to a newly determined O_A after its transmission, node A updates $O_{max,A}$ when the transmission by node B ends and this procedure is repeated until the transmission order of node

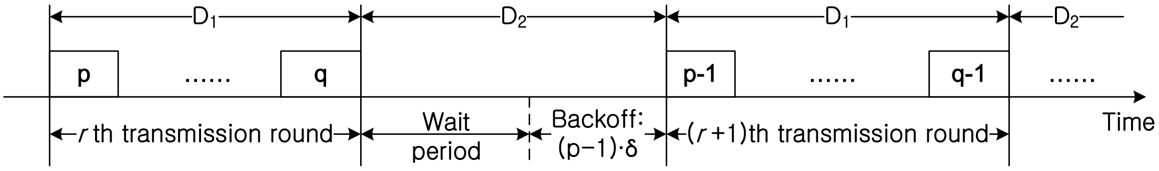


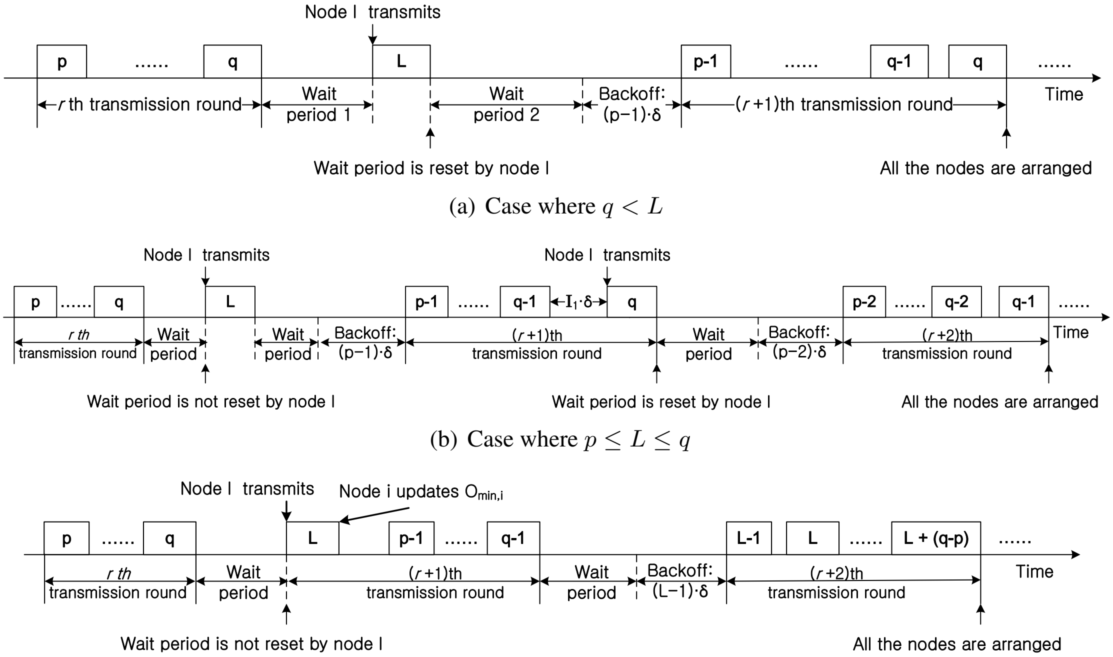
Figure 3.5 Description of time durations D_1 and D_2.

A becomes zero at which time node A changes O_A to $m_A + 1 = 2$. The transmission orders of nodes A and B are switched with each other after the end of each transmission round. Figure 3.4 shows how nodes A and B transmit in a round robin manner.

Operation of TOD-MAC when a node newly joins or leaves the network

Let us now look at how TOD-MAC operates when a node newly joins or leaves a network. Let O_{min} and O_{max} be p and q $(> p)$ in the rth transmission round, respectively, and assume that transmission orders are well arranged from p to q without overlap. First, let us consider the case when node l newly joins the network, and randomly selects its transmission order O_l from $[1, CW_l]$, and sets BC_l to O_l and WP_l to $n_w \cdot \delta$. Node l may then transmit in the middle of the transmission round (D_1) or between the two consecutive transmission rounds (D_2), described in Fig. 3.5. Here, the case where node l transmits during D_1, which would result in a collision, is not considered (the collision handling method will be described in the next section). When node l transmits during D_2 after the rth transmission round, there are three possible cases depending on the value of O_l, which node l has chosen from $[1, CW_l]$ as its transmission order $(O_l = L)$ in the rth transmission round; i.e., 1) $q < L$, 2) $p \leq L \leq q$ and 3) $L < p$.

Figure 3.6(a) shows how each node arranges its transmission order for the case where $q < L$. After the transmission by node l, node i, $i \neq l$, updates $O_{max,i}$ once more because $O_{max,i} = q < L$, and consequently resets BC_i to O_i and WP_i to $n_w \cdot \delta$ in accordance with Table 3.3. In the same time, node l calculates its transmission order $O_l = O_{min,l} + m_l = p + (q - p) = q$ for the



(a) Case where $q < L$

(b) Case where $p \leq L \leq q$

(c) Case where $L < p$

Figure 3.6 Operation of TOD-MAC when node l newly joins the network

$(r + 1)$th transmission round, and sets BC_l to O_l and WP_l to $n_w \cdot \delta$. Note that the transmission order of node i in the $(r + 1)$th transmission round is decreased by one after its transmission, in accordance with (3.7). Therefore, node l becomes node $n(max)$ and the transmission order of each node can be arranged in the $(r + 1)$th transmission round as can be seen in Fig. 3.6(a).

In the case where $p \leq L \leq q$, the transmission order of each node can be arranged in the similar way as described in the above except that the transmission order of each node is completely arranged in the $(r+2)$th round instead of the $(r+1)$th round. When node l transmits a data frame, node i, for $i \neq l$, does not updates $O_{min,i}$ or $O_{max,i}$ because $O_{min,i} = p \leq L \leq q = O_{max,i}$, and so it simply increases m_i by one. On the other hand, node l calculates its transmission order O_l as $O_{min,l} + m_l = p + (q - p) = q$ in the $(r + 1)$th round, and sets BC_l to O_l and WP_l to $n_w \cdot \delta$. Note that $(WP_l + BC_l) - (WP_i + BC_i) > 1$, $i \neq l$, because $WP_i < WP_l = n_w$ and $BC_i < BC_l = q$. Thus, although node l becomes $n(max)$ in the $(r + 1)$th round, the number of idle slots (I_1) between the transmissions of node l and node $n(q - 1)$ is greater than one. When node l transmits in the $(r + 1)$th round, node i updates $O_{max,i}$ to q and resets WP_i to $n_w \cdot \delta$, and node l also sets WP_l to $n_w \cdot \delta$. Therefore, the transmission order of each node can be properly arranged in the $(r + 2)$th round, as can be seen in Fig. 3.6(b)

Figure 3.6(c) shows how each node arranges its transmission order for the case where $L < p$. After the transmission by node l, node i updates $O_{min,i} = p$ $(> L)$ to L and notices that the $(r + 1)$th round has started and O_{min} has changed to L. Consequently, it calculates O_i as $O_{min,i} + m_i = L + m_i$ after its transmission in the $(r + 1)$th round, and O_i is between L and $L + (p - q)$ in the $(r + 2)$th transmission round. Note that node l simply decreases O_l to $L - 1$ in the $(r + 2)$th round because it is $n(min)$ in the $(r + 1)$th round. Therefore, the transmission order of each node is arranged from $L - 1$ to $L + (p - q)$ in the $(r + 2)$th transmission round.

Let us now consider the case when node l leaves the network in the rth round. Note that

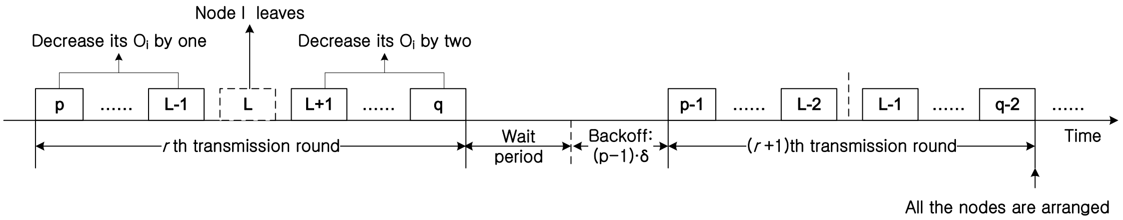


Figure 3.7 Operation of TOD-MAC when a node l newly leaves the network

L must be between p and q. The transmission order of node $n(k_1)$, where $p \leq k_1 < L$, is not affected when node l leaves the network, and is decreased by one, in accordance with (3.7) in the $(r+1)$th round. On the other hand, node $n(k_2)$, where $L < k_2 \leq q$, sets its transmission order O_{k_2} to $O_{min,k_2} + m_{k_2} = p + (k_2 - p - 2) = k_2 - 2$; i.e., it decreases its transmission order by two because node l has left the network. Therefore, the transmission order in the network is well arranged from $p-1$ to $q-2$ in the $(r+1)$th round, as can be seen in Fig. 3.7.

3.3.3 Handling Collision and Unsaturated nodes

In this section, we look at how transmission collisions are handled by each node. In TOD-MAC, a receiver always sends an ACK frame to the sender except when the transmitted data frame header has been corrupted by channel noise. We assume that the data rate is set sufficiently low so that data frame headers are not corrupted by channel noise. Thus, each node can easily know whether a transmission fails because of collisions or channel errors using the ACK frame from the receiver. Hereafter, we will call a node that fails to transmit a frame because of collision a $collision$ node, and a node that successfully transmits a frame a $normal$ node.

After a transmission failure, collision node i doubles CW_i, using (3.8), and randomly selects its transmission order O_i from the range $[1, CW_i]$ with the newly determined CW_i.

$$CW_i = \begin{cases} CW_i * 2, & \text{if } CW_i < CW_{max}, \\ CW_{max}, & \text{otherwise.} \end{cases} \tag{3.8}$$

After determining the value of O_i, collision node i sets BC_i to O_i. Unlike the other nodes that successfully transmit a frame, 1) it sets $O_{min,i}$ and $O_{max,i}$ to 128 and zero, respectively, and 2) the WP_i value of collision node i is kept at zero until there is a successful transmission. The reason underlying this is as follows. It is noted that the newly selected transmission order O_i can be less than O_{min} or greater than O_{max}. When O_i is less than O_{min} and collision node i sets $O_{min,i}$ to O_i, it does not update $O_{min,i}$ after the transmission of $n(min)$ because $O_{min,i} < O_{min}$. Consequently, it cannot determine when the $next$ round starts from the value of $O_{min,i}$. Likewise, when O_i is greater than O_{max} and collision node i sets $O_{max,i}$ to O_i, it does not update $O_{max,i}$ after the transmission of $n(max)$ because $O_{max,i} < O_{max}$. Consequently, it cannot determine when the $current$ round ends from the value of $O_{max,i}$. Thus, it sets $O_{min,i}$ and $O_{max,i}$ to 128 and zero, respectively, so that it can correctly determine these when the $next$ round starts and when the $current$ round ends. Let us now look at why collision node i has to be kept WP_i at zero after its transmission fails. If a collision node retransmits right after a collision in the current transmission round, there is a high probability that the collision node will experience another collision because the idle time between the two consecutive transmissions is a single slot. In order to avoid another collision, collision node i decreases BC_i only for idle slots in the wait period, and consequently transmits data only during the wait period. In TOD-MAC, the number of idle slots I between two consecutive transmissions is one during the transmission round and can be larger than one only during the wait period (see Fig. 3.1(a)) or at the time of network initialization. After the first round, with the exception of collision nodes, each node knows its transmission order. Therefore, collision nodes can deduce with a high probability that the wait period has started when I is larger than one after the first transmission round. In this paper, collision nodes i judges that the wait period has started and decreases BC_i when I is larger than or equal to three and WP_i is larger than zero. The parameter update for collision node i in the event of idle slot (i.e., backoff procedure) is described in Table 3.2.

Thus, if WP_i of collision node i remains to zero, it does not retransmit a frame until there is a successful transmission from other nodes. By not retransmitting a frame right after its collision, it can avoid other possible collisions in the current transmission round. Table 3.3 summarizes the method by which node parameters are updated in the event of a collision.

Note that the possibility exists for all the nodes in a network to become collision nodes and consequently set their WPs to zero. if such a possibility occurs, no node can perform the backoff procedure because a node can transmit only after a successful transmission. As a result, there would be no activity in the network until a node newly joins and successfully transmits a frame. In order to prevent such a scenario, collision node i decreases BC_i when I is larger than $2 \cdot n_w \cdot \delta$ even if WP_i remains at zero.

After collision node i updates parameters CW_i, O_i, $O_{min,i}$, and $O_{max,i}$ as described above, it operates like the other normal nodes, except that it decreases its backoff counter only during the wait period. If collision node i successfully transmits a frame in the wait period, it then determines its transmission order in accordance with (3.7) and becomes a normal node. Otherwise, it doubles CW_i once more and starts the backoff procedure only during the wait period until it successfully transmits a frame. If a collision occurs, the other nodes j, $j = 1, 2, \ldots, N$, $j \neq i$, that are not affected by the collision do nothing (do not update O_j, $O_{min,j}$, $O_{max,j}$, and WP_j).

Figure 3.8 shows in which a collision occurs in a network comprising three nodes, A, B, and C. We assume that each node sets WP_i to $10 \cdot \delta$ (i.e., $n_w = 10$) and nodes A, B, and C randomly set their transmission orders to three, five, and five, respectively. In the first round, transmissions by nodes B and C fail due to collisions, and only node A successfully transmits a data frame. After the transmission, node A determines O_A to two, in accordance with (3.7), and sets $O_{max,A}$, $O_{min,A}$, and BC_A to 2, and resets WP_A to $10 \cdot \delta$. When the collision occurs, node A does nothing, whereas nodes B and C double their CWs and randomly choose their

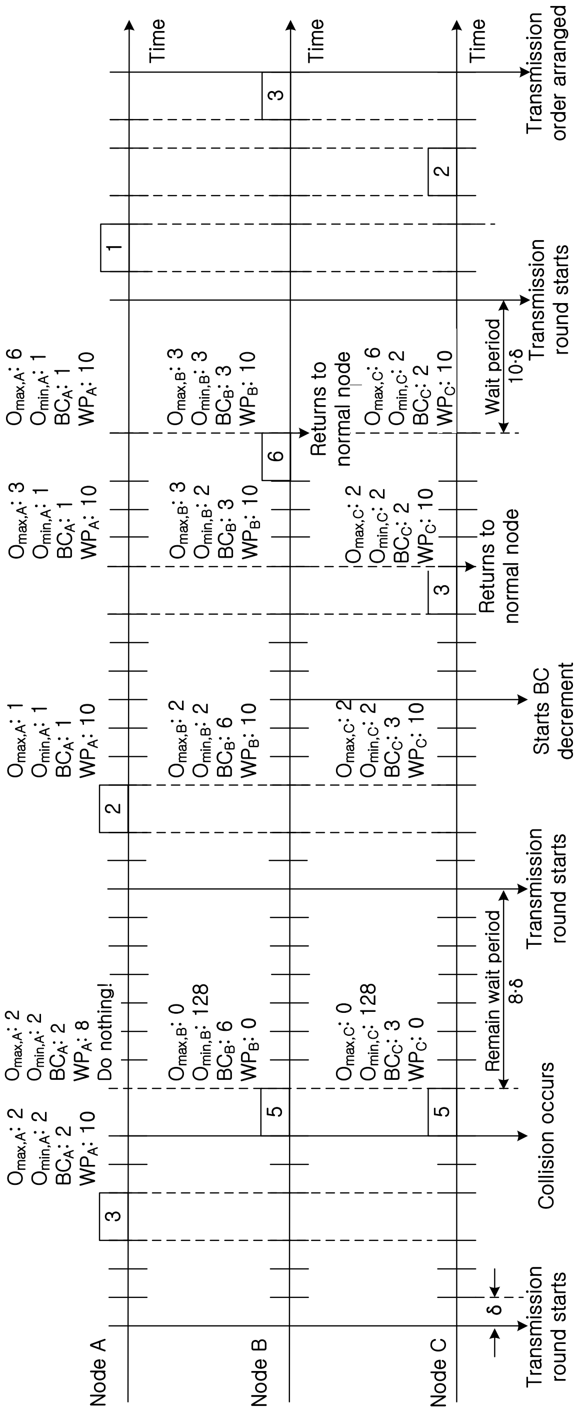


Figure 3.8 Collision handling in TOD-MAC

transmission orders from the newly determined CWs. Let us assume that nodes B and C set their transmission orders to six and three, respectively. Nodes B and C then, respectively, set $O_{max,B}$ and $O_{max,C}$ to zero, $O_{min,B}$ and $O_{min,C}$ to 128, and keep WP_B and WP_C at zero. Note that they cannot start to decrease their BCs in the first wait period even if I is larger than three because their WPs are zero. After $10 \cdot \delta$ ($= WP_A + BC_A$, presently WP_A is $8 \cdot \delta$) has elapsed after the collision, node A, which is node $n(min)$ with $O_A = 2$ in the second round, transmits a data frame. Nodes B and C then update their $O_{max,B}$, $O_{min,B}$, $O_{max,C}$ and $O_{min,C}$ to two. Note that the WPs of nodes B and C are initialized to $10 \cdot \delta$ because $O_{max,B}$ and $O_{max,C}$ are updated. The second wait period starts at the end of the transmission by node A. In the second wait period, nodes B and C start the backoff procedure because their WPs are larger than zero. They wait I is larger than three and start to decrease their BCs during the wait period. Node C transmits when BC_C becomes zero and sets O_C to $O_{min,C} + m_C = 2 + 0 = 2$, and then operates like a normal node. After three more idle slots have passed, node B also successfully transmits a data frame and sets O_B to $O_{min,B} + m_B = 2 + 1 = 3$. In the third round, the transmission orders are properly arranged and nodes A, B, and C transmit in a round robin manner.

An unsaturated node, whose aggregated length of packets in the queue is smaller than the required frame body length (S_{fb}), cannot correctly inform others of its transmission order using the transmission duration. When a node cannot make a frame of the required length S_{fb} because of lack of packets in the queue, it does not transmit during the transmission round and transmits only in the wait period. That is, an unsaturated node decreases BC_i when I is equal to or greater than three and WP_i is larger than zero in accordance with Table 3.2. An unsaturated node i operates like a normal node when it can make a frame of required length S_{fb}. If unsaturated node i fails to transmit a frame because of collisions in the wait period, it doubles CW_i, using (3.8), and selects BC_i from the newly determined CW_i. Otherwise, it sets CW_i to CW_{min} and

Table 3.4 System parameters used in the simulations.

Subsection	3.4.1			3.4.2 and 3.4.4			3.4.3		
Scheme	TOD	AFR	DCF	TOD	AFR	DCF	TOD	AFR	DCF
Data rate (Mbps)	varied	varied	varied	65	65	65	H: 65 L: 6.5	H: 65 L: 6.5	H: 65 L: 6.5
Basic rate (Mbps)	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5
CW_{min}	32	32	32	32	32	32	32	32	32
CW_{max}	256	1024	1024	256	1024	1024	256	1024	1024
Frame Size (B)	8192, 16384	8192, 16384	8192, 16384	10240	10240	10240	H: 10240 L: 1024	H: 10240 L: 1024	H: 10240 L: 10240
Packet Size (B)	2048	2048	2048	2048	2048	2048	2048	2048	2048
Fragment Size (B)	256	256	-	256	256	-	256	256	-

randomly chooses BC_i from $[1, CW_i]$. If each node in a network is unsaturated, TOD-MAC operates in much the same fashion as IEEE 802.11 DCF; i.e., all the nodes contend with each other using the binary exponential backoff, because there is no transmission round and only the wait periods are concatenated one after the other. The only difference is that in TOD-MAC each node has to wait three additional idle slots before decreasing its backoff counter. By doing this, TOD-MAC can effectively deal with transmission collisions as well as unsaturated nodes.

3.4 Simulation and performance evaluation

In this section, we report the result of performance evaluation conducted on TOD-MAC using an NS2 simulator, and how they compared to those of the AFR scheme [4] and IEEE 802.11n DCF with packet aggregation and Block ACK. The system parameters used in the simulations are listed in Table 3.4. H and L in the experiment of Subsection 3.4.3 denote the high and low rates in multi-rate network, respectively.

Because the frame size varies according to the transmission order in TOD-MAC, we give only the average frame size in TOD-MAC. Frame size is fixed in the AFR scheme and 802.11n DCF. In order for all the schemes to have the same payload size on average, we set T_{BTD} in TOD-MAC as

$$T_{BTD} \quad = \quad \frac{(S_f + S_{hdr}^{phy} + S_{hdr}^{mac}) \cdot 8}{R} - \frac{\Sigma_{i=1}^{N} \lambda * i}{N}. \tag{3.9}$$

Here, S_f represents the payload size of the AFR scheme and 802.11n DCF.

As the value of λ in TOD-MAC increases, the probability that a node becomes an unsaturated node increases and MAC efficiency can decrease. Therefore, we set λ to 4μs, because it is the minimum time required for packet detection/clear channel assessment (CCA) in 802.11n [11, 12]. We also set n_w to six, i.e., we set node i, $i = 1, 2, \ldots, N$, to reset WP_i to $6 \cdot \delta$ when it updates $O_{max,i}$ or the transmission of node i ends. (When the rate of packet generation is not explicitly mentioned, it is assumed that each node always has enough packets to transmit.) Nodes were randomly placed in a rectangular area 100 m by 100 m. The packets were sent without RTS/CTS. We used MAC efficiency, throughput, collision rate, and air-time fairness as performance measures.

3.4.1 MAC efficiency performance

Figure 3.9 shows the MAC efficiency ($\frac{Throughput}{PHYrate} \cdot 100\%$) for various payload sizes (8192 and 16384 B) and PHY rates (from 65 Mbps to 585 Mbps) for a network comprising 10 nodes. Although, in the figure, the MAC efficiency of TOD-MAC decreases as the PHY rate increases, it can be seen that it was always greater than 50% regardless of the payload size or PHY rate. Furthermore, TOD-MAC provided the best MAC efficiency compared to the ARF and DCF, at approximately 20% higher. Since high throughput performance implies high MAC efficiency

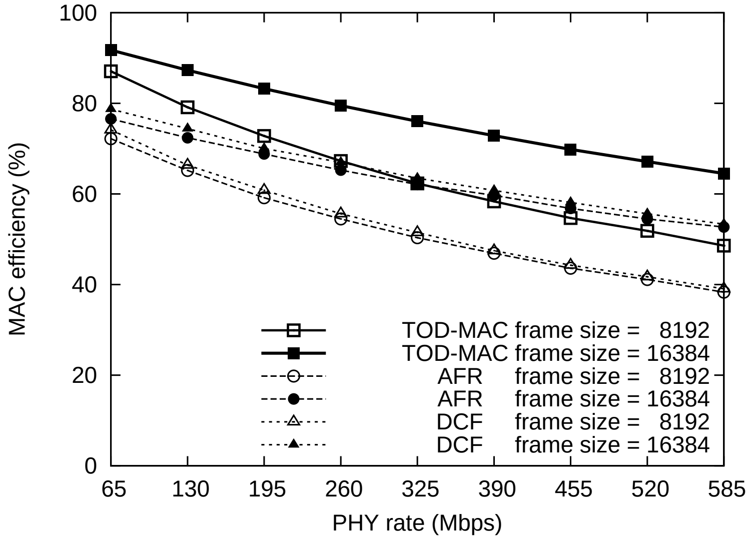


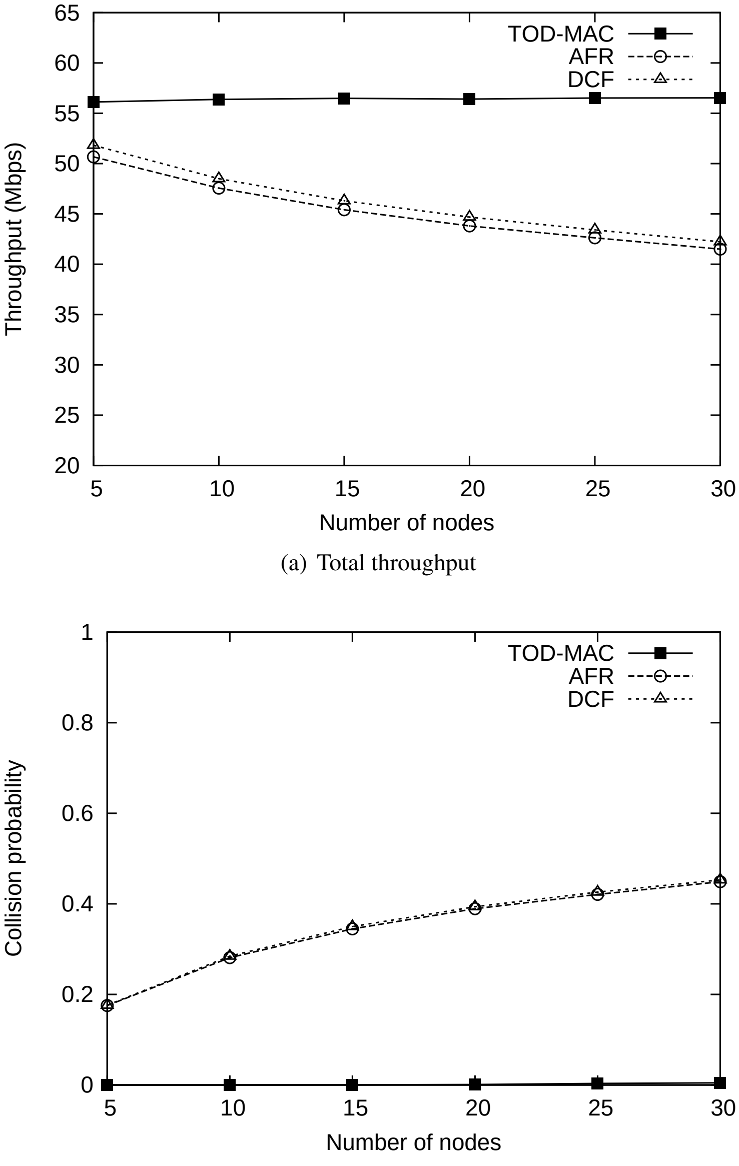
Figure 3.9 MAC efficiency for various payload sizes with increasing PHY rates for a network comprising ten nodes

for a fixed PHY rate, we present the throughput performance instead of the MAC efficiency in the following subsections.

3.4.2 Single rate network

We investigated the total throughput and collision rate for a 65 Mbps network. Figure 4.8(a) shows the total throughput for various numbers of nodes in the network. Regardless of the number of nodes, TOD-MAC maintained a relatively constant throughput, which was $10\% - 35\%$ higher than those of the AFR scheme and DCF. This is because the collision rate of TOD-MAC is virtually zero (see Fig. 3.10(b)).

Figure 3.11 shows the throughput performance for various numbers of nodes in a noisy channel. In the noisy channel of $BER = 10^{-4}$, the throughput performance of DCF was severely degraded because packets that were aggregated to form a frame 2048 B long, which was longer than the fragment sizes in the AFR scheme and TOD-MAC. TOD-MAC also gave the best throughput performance for BERs of 10^{-4} and 10^{-5}.



(a) Total throughput

(b) Collision rate

Figure 3.10 Total throughput and collision rate in a single transmission rate network

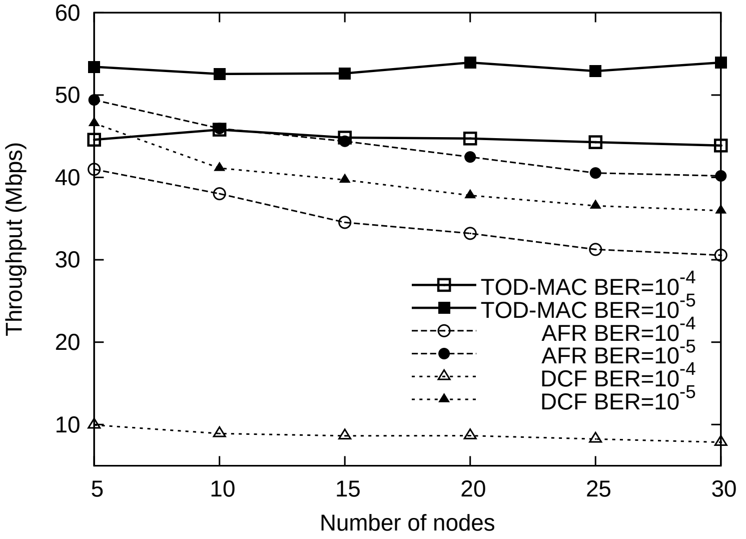


Figure 3.11 Throughput performance in a channel error environment

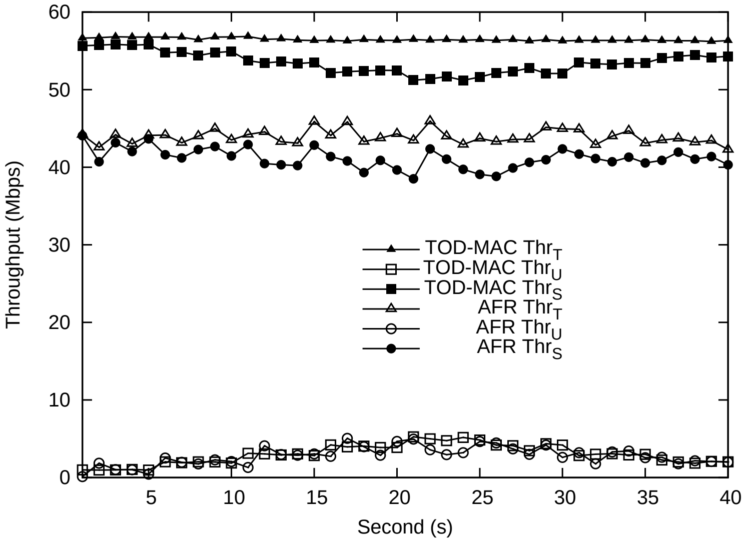


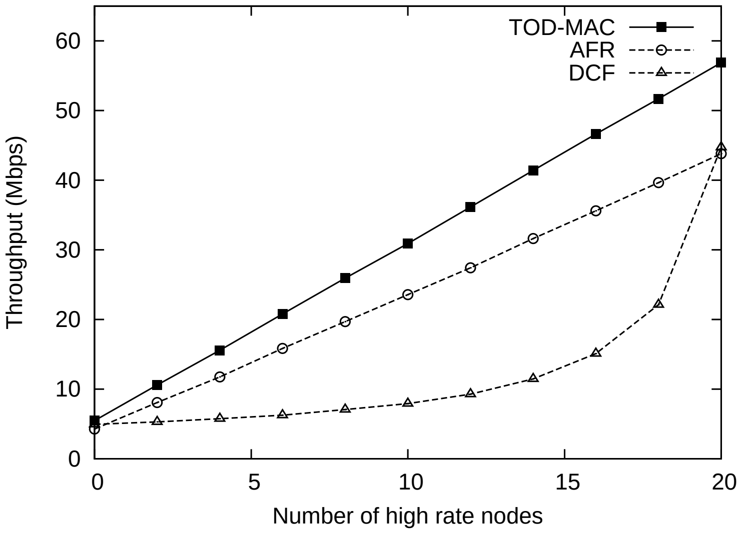
Figure 3.12 Throughput performance for 10 saturated nodes and one unsaturated node

Figure 3.12 shows the throughput performance when there were 10 saturated nodes and one unsaturated node in the network. In the figure, Thr_T, Thr_U, and Thr_S represent the total throughput, aggregate throughputs of the unsaturated node and saturated nodes, respectively. The offered load of the unsaturated node was varied every 5 s as $(1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1)$ Mbps. Note that the unsaturated node became a saturated node when the offered load was larger than 6 Mbps. In TOD-MAC, the unsaturated node used a sufficient amount of channel bandwidth to transmit its offered load, and the saturated nodes effectively shared the remaining bandwidth of the unsaturated node. The total throughput was approximately 25% higher in TOD-MAC compared to the AFR scheme at all times. Even though we also studied the case for IEEE 802.11n DCF, its simulation results are not presented in Fig. 3.12 because they were very similar to those of the AFR scheme. (Further, including them would make the graph difficult to comprehend.)

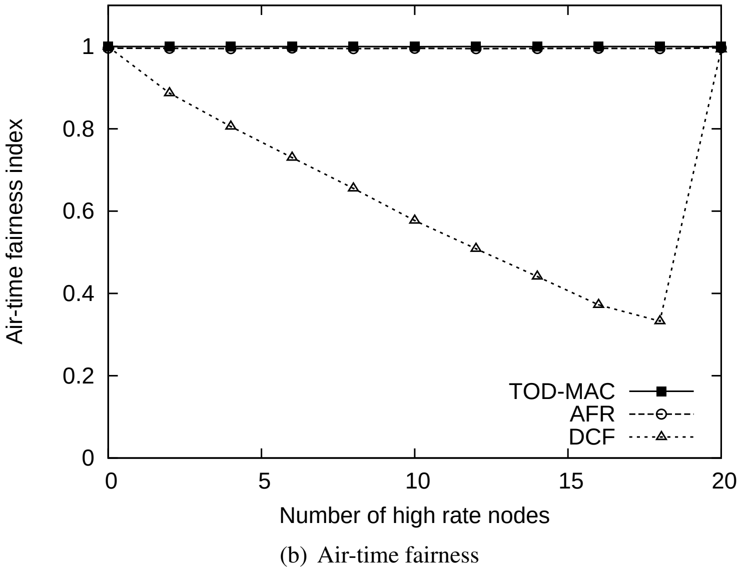
We confirmed that TOD-MAC has the best performance in single rate networks under various environments because it is based on a round robin algorithm that does not use control messages.

3.4.3 Multi-rate network

We also evaluated the throughput performance and Jain's fairness index of air-time in a multi-rate network. We considered a scenario in which there are 20 senders in a network and the number of high rate nodes varied from zero to 20 (i.e., the number of low rate nodes varied from 20 to zero). The data rate of high and low rate nodes were 65 and 6.5 Mbps, respectively. Since frame size is adjusted according to the data rate in TOD-MAC and the AFR scheme, the throughput performance increased linearly as the number of high rate nodes increased, as can be seen in Fig 3.13(a). In contrast, IEEE 802.11n DCF had a poor throughput performance because it did not adjust the frame size. Figure 3.13(b) shows that TOD-MAC and the AFR



(a) Throughput performance



(b) Air-time fairness

Figure 3.13 Throughput performance and air-time fairness in multi-rate network

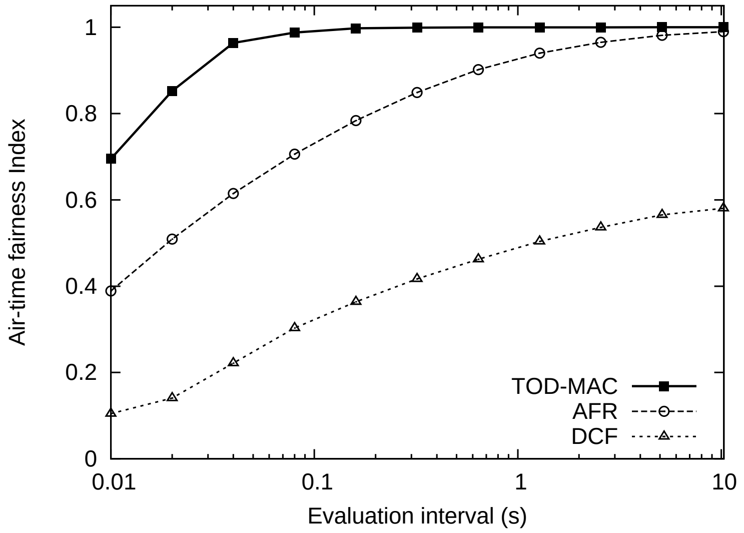


Figure 3.14 Short-term air-time fairness in multi-rate network

scheme achieved excellent air-time fairness for various numbers of high rate nodes in the multi-rate network.

Figure 3.14 shows the short-term air-time fairness when there were five high rate nodes and five low rate nodes in the network. TOD-MAC gave the best short-term air-time fairness compared to the AFR scheme and DCF because it is based on a round robin scheme and the transmission duration for each node is nearly the same on average. In TOD-MAC, the air-time was fairly allocated to each node at the end of the tenth round ($10 * T_{send}^{Avr} * 10 = 0.1322$ s), as can be seen in Fig. 3.14. This shows that TOD-MAC also achieved the best throughput performance and air-time fairness in multi-rate network as in the single rate network.

3.4.4 Transient time behavior

Finally, we investigated how long it took for the transmission order in a network to stabilize when all the nodes newly joined the network and randomly selected their transmission orders. For this, we simulated a scenario in which 20 nodes randomly selected their transmission order

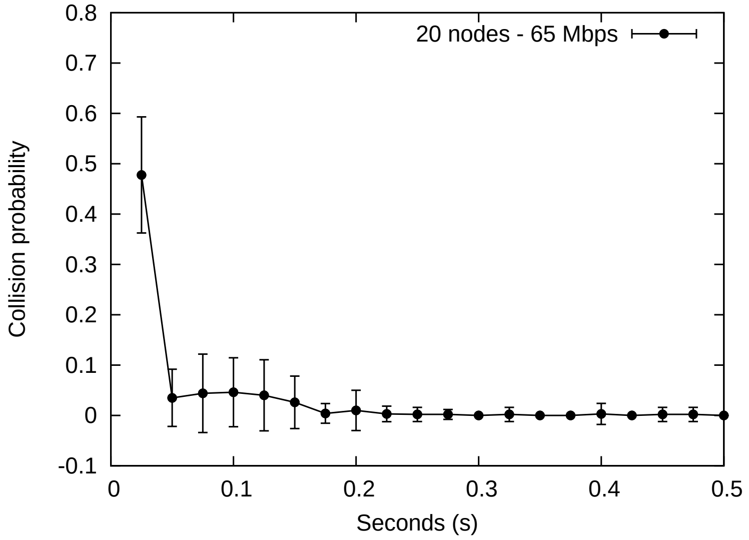


Figure 3.15 Transient behavior of collision rate

and measured the variation in collision rate from 100 simulation results. In Figure 3.15, the circle and error bars indicate the average collision rate and its standard deviation, respectively. It took approximately 0.1 s for the average collision rate to become nearly zero, and approximately 0.2 s for the average collision rate and its standard deviation to stabilize.

Figure 3.16 shows the aggregate and per-node throughputs when five nodes joined a network of five nodes at 5.0 s, and then left the network at 10.0 s. In TOD-MAC, the throughput performance stabilized quickly as soon as the nodes joined or left the network. This simulation result confirmed that TOD-MAC can achieve high throughput performance within a very short transient time despite the changes in the number of nodes in a network.

3.5 Chapter summary

In this paper, we proposed the TOD-MAC that enhances MAC efficiency in WLANs. Unlike traditional MAC protocols, in TOD-MAC, the transmission duration is adjusted and it performs the function of a control message to determine the transmission order of nodes. Based on infor-

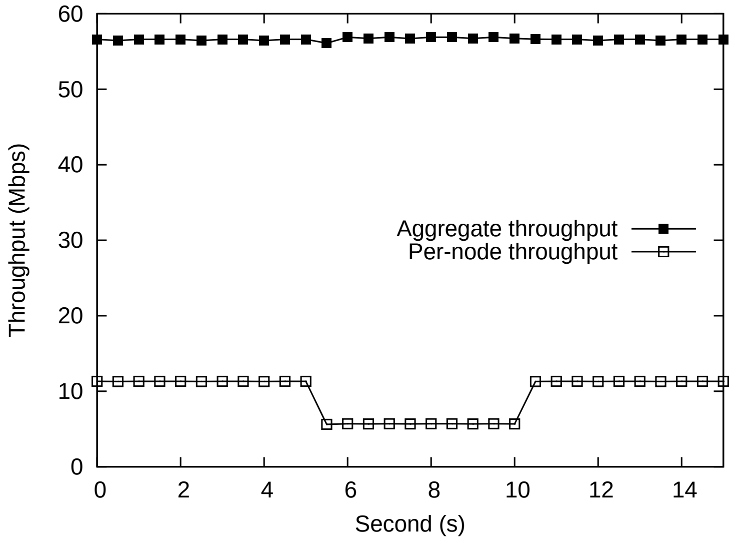


Figure 3.16 Transient time behavior when five nodes newly joined and then left the network

mation of transmission order, each node transmits in a round robin manner, which minimizes the idle time between two consecutive transmissions and also prevents transmission collisions. Consequently, TOD-MAC achieves high throughput performance in various simulation scenarios. Furthermore, the simulation results show that it provides good short/long term air-time fairness, and fast transient response in dynamic environments. TOD-MAC can operate not only in WLANs, but also in any CSMA/CA networks. In addition, the basic principle underlying TOD-MAC is simple, which makes it easy to implement.

Chapter 4

Improving Max-min air-time fairness in IEEE 802.11n ad-hoc networks

4.1 Background and Related Work

The recent explosive proliferation of mobile devices such as smart phones and tablet PCs accelerates the demand for wireless Internet access. The wireless local area network (WLAN) is one of the most popular wireless communication technology thanks to its ease of deployment and low installation cost. At the same time, the demand for services based on ad-hoc networks rapidly increases in WLANs. Providing fair service among nodes is important in the operation of ad-hoc networks. When a user does not get a fair service compared to other users, he/she will not be satisfied with the wireless service provider.

The principle for a MAC protocol to achieve fairness is simple, i.e., to adjust the contention window (CW) of a node according to its current share of channel resources. However, designing a protocol that works well in ad-hoc networks is very difficult and has not yet been successful. This is due to the combination of several factors such as the physical nature of wireless communication, random access mechanism to wireless channel, arbitrary spatial distribution of nodes, multiple objectives in the operation of an ad-hoc network. Some of the objectives is to make nodes share the wireless channel fairly with their neighbors and at the same time maximize the network throughput with small transmission delay. These objectives are usually conflicting with each other and one has to find a suitable compromise among these objectives.

Extensive studies have been carried out to find an optimal MAC, and one of them is to achieve max-min fairness [28–38]. The notion of max-min fairness was introduced for wired networks [39], and it cannot be directly applied to wireless networks. To achieve max-min fairness in a wireless network, a node can use additional channel resource as long as it does not take the resource away from the others who use less channel resource. (The channel resource can be throughput, air-time or the number of transmissions, etc. In this dissertation, we will consider air-time as the channel resource that should be fairly utilized and elaborate on the definition of max-min air-time fairness later.) In general, if a node uses more channel resource, other nodes have to use less channel resource when channel resource are limited. Let us consider a node attempting to use more channel resource to increase its air-time. If the other nodes can still maintain the same air-time as before, then these nodes have nothing to complain about this attempt, and this improves the overall channel utilization. Otherwise, they will complain about this attempt, because channel utilization becomes less fair. This is the reason why it is important to achieve max-min fairness.

It is well known that the DCF protocol does not provide fair throughput to the nodes in ad-hoc networks [40]. A flow contention graph [28, 29] was proposed to calculated the max-min fair share of a node, up to which a node can utilize wireless channel resource. However, each node must know the topology of its neighbor nodes to obtain the flow contention graph, which may severely degrade the MAC efficiency due to the heavy traffic caused by control packets. Furthermore, these schemes show poor transient response in dynamic environments.

Some schemes [32,33] tried to improve max-min fairness by making heavy users yield transmission opportunities to other users, but these did not necessarily provide max-min fairness [38]. In [34–37], each node uses the channel resource in accordance with the weight that is calculated from its fair share. However, theses schemes are not practical to use in ad-hoc networks, because it is difficult to calculate the fair share of a node in a distributed way and so the weights of nodes are unknown in general. In the Distributed Filling and Draining (DFD) scheme [38], each node periodically broadcasts its throughput potential (THP) information using a control frame, and adjusts its CW_{min} and CW_{max} according to the THP information announced by other nodes. Although the DFD scheme can improve max-min throughput fairness in ad-hoc networks, it requires additional control messages as in [28, 29]. This induces additional overhead that can severely degrade MAC efficiency in high speed WLANs because control messages are transmitted at the basic rate as the PHY header. Moreover, it cannot be guaranteed that the schemes based on the control frames improve max-min throughput fairness among the nodes that are in carrier sensing range but out of transmission range of each other, because they may not obtain necessary information from the control frame. Recently, Douglas $et\ al.$ [41] exploited the transmission opportunity (TXOP) feature for transmitting burst of packets in the IEEE 802.11e/n MAC, to enhance max-min throughput fairness in mesh networks formed from a set of inter-connected WLANs that are non-interfering, which is not suitable for general ad-hoc networks.

In order to achieve max-min fairness, it is very important to take the hidden node problem into consideration, because it is highly probable that there are hidden nodes in an ad-hoc network and the fairness can severely degraded in such a case [6, 7]. Several analytic models were derived to estimate the impact of hidden nodes on network performance for simple network topologies [42–44]. The process to alleviate the effects of hidden nodes can be accomplished into two steps: (1) hidden node detection [8,45–47] and (2) hidden node resolution [9,48–52].

In the hidden node detection scheme of [45], a sender deduces that there are hidden nodes in the networks when only an ACK frame is observed without observing the corresponding data transmission. This method can be successfully applied in a single BSS network, but does not work properly in general ad-hoc networks. In another hidden node detection mechanism, nodes cooperate with access points (APs) in a densely deployed network [46]. Because the cooperation with APs is essential for hidden node detection, it cannot be used in general ad-hoc networks. Kim $et\ al.$ [8] proposed the hidden node detection scheme that exploits the new features of IEEE 802.11n such as the frame aggregation and block ACK. This scheme can detects hidden nodes quite well in IEEE 802.11n network environment [8], and we will incorporate this scheme into the MAC protocol to improve max-min fairness in Section 4.2.

Aside from hidden node detection schemes, several hidden node resolution schemes were proposed [9, 48–52]. Exchanging RTS/CTS control frame, which is probably the most well known hidden node resolution scheme in WLANs, was introduced in the multiple access with collision avoidance (MACA) [48]. Moreover, several Receiver-oriented Contention (ROC) schemes [49–51] were designed based on the MACA scheme to suppress the transmission of RTS frame at the sender. The ROC schemes show better performance compared to the RTS/CTS exchange scheme when there are hidden nodes [49]. A dual busy tone multiple access (DBTMA) mechanism [52] was proposed to alleviate the hidden node problem by using busy tone signal to prevent the interference from hidden nodes during data transmission. However, it does not use ACK frames, and requires additional transceivers, control and data sub-channels with sufficient spectral separation to avoid inter-channel interference, and thus it is not compatible with the IEEE 802.11 standard. Kim [9] proposed a method to alleviate the hidden node problem by extending the effective CTS range and adaptively adopting the ROC mechanism. In this method, a node can identify a CTS frame that is sent out of the transmission range but within the carrier sensing range. After identifying the CTS frame, the node can appropriately set its network allocation vector (NAV) value with the help of the frame size adaptation scheme. In addition, the ROC mechanism is initiated only when interference from hidden nodes is detected via the hidden node detection mechanism [8]. In this way, this scheme can effectively resolve the hidden node problem for various network topologies. However, it does not take fairness into consideration and its fairness performance can be poor in the presence of the starvation problem as in Fig. 1.2.

In this chapter, we propose a max-min air-time fairness MAC (MAF-MAC) for improving max-min air-time fairness in IEEE 802.11n ad-hoc networks. In traditional MAC protocols, the length of a packet or a frame is usually fixed and the transmission duration is determined by the data rate. If packets can be divided into small fragments, the aggregated frame size can be adjusted at the fragment level, and the transmission duration can also be controlled so that it can play the role that is usually carried out by control messages. This idea is simple, but very effective to achieve max-min air-time fairness in ad-hoc networks without incurring any overhead. In MAF-MAC, a node estimates the ratio of its air-time usage with respect to the total channel busy time, and announces this ratio to the nodes in its carrier sensing range. Each node adjusts its CW value based on this information to improve max-min air-time fairness. Furthermore, MAF-MAC adopts the hidden node detection [8] and resolution mechanism [9], and thus it can provide good air-time fairness while effectively utilizing the channel even when there are hidden nodes in a network. Moreover, MAF-MAC can automatically resolve the performance anomaly in multi-rate networks, where high rate and low rate nodes achieve the same throughput in IEEE 802.11 DCF that provides the same transmission opportunity to each node [53]. Because of this, a low rate node occupies the channel longer than a high rate node, which is good with respect to the throughput fairness performance, but can significantly deteriorate the overall network throughput. MAF-MAC naturally takes care of this problem even in a general ad-hoc networks, not like the methods for a single cell network [54–58].

The rest of this chapter is organized as follows. Section 4.2 explains the operation of MAF-MAC that improves max-min air-time fairness in ad-hoc networks. In Section 4.3, we discuss how to handle the issues such as hidden nodes, transmission collision and overlap, the presence of unsaturated nodes among saturated nodes, and enhancing channel utilization when wireless channel is underutilized in implementing MAF-MAC. Section 4.4 evaluates the performance of MAF-MAC and compares it to other MAC schemes in very high-speed WLANs [1, 8, 9]. Section 4.5 concludes this chapter.

4.2 Max-min Air-time Fair MAC (MAF-MAC) protocol in ad-hoc networks

In this section, we explain how MAF-MAC works to improve max-min air-time fairness in an ad-hoc network. In IEEE 802.11 ad-hoc networks, node i shares channel resource with its neighbor nodes that are in the carrier sensing range of node i. Let us define a neighbor node of node i as the following.

Definition 4.1 (neighbor node)**.** *Node j is a neighbor node of node i if it is in the carrier sensing range of node i.*

In the following, we denote the set consisting of node i and its neighbor nodes as S_i and the number elements in S_i as N_i.

Max-min fair was originally defined for wired networks [39], where transmission rate r_i is allocated for session s_i. The rate vector $\overrightarrow{r} = (r_1, r_2, \ldots, r_N)^T$ is said to be feasible if the sum of rates for sessions crossing each link in a network does not exceed the link capacity, where N is the number of nodes in the network. Then, the max-min fair is defined as the following.

Definition 4.2 (max-min fair [39])**.** *A rate vector $\overrightarrow{r}$ is said to be max-min fair if it is feasible and it is impossible to increase the rate of a session without losing feasibility or decreasing the* *rate of another session that has a smaller rate.*

The sessions crossing a link share the link capacity in wired networks, whereas a node shares wireless channel with its neighbor nodes in ad-hoc networks. It is difficult to test the feasibility of a given rate allocation for nodes because nodes access the channel in a distributed manner and channel resource can be spatially reused. Therefore, in order to further discuss MAC protocols to improve max-min fairness in ad-hoc networks, a measure needs to be defined to show how much channel resource a node uses. It can be throughput, air-time, the number of transmissions or the time spent for successful transmission in a given time interval. In this dissertation, we use air-time as a fairness measure and a MAC protocol to improve fairness with respect to this measure is proposed, but other measures can be used similarly. Let A_i be the channel access time of node i in time interval T, and $a_i = A_i/T$ be the normalized air-time of node i. We will simply call a_i as air-time of node i if no confusion arises. Then, we can easily see that the air-time vector $\overrightarrow{a} = (a_1, a_2, \ldots, a_N)^T$ is feasible if $\sum_{k \in S_i} a_k \leq 1$, for all i. (T is assumed be fixed for ease of explanation.) Similar to Definition 4.2, the max-min air-time fair is defined as the following.

Definition 4.3 (max-min air-time fair in wireless network)**.** *An air-time vector $\overrightarrow{a}$ is said to be max-min fair if $\sum_{k \in S_i} a_k \leq 1$, for all i, and it is impossible to increase the air-time of a node without decreasing the air-time of another node that has a smaller air-time.*

Jain's fairness index, which is usually used for a single cell network to measure its fairness performance, is not appropriate to evaluate the fairness performance for an ad-hoc network consisting of multiple cells. This is because each node has different neighbor nodes, and the number of neighbor nodes generally differs from one node to another. Thus, we define a new fairness index to evaluate max-min air-time fairness for ad-hoc networks.

Definition 4.4 (Jain's air-time fairness index for node i and generalized Jain's air-time fairness index). *Jain's air-time fairness index for node i is defined as*

$$J_i = \frac{(\sum_{k \in S_i} a_k)^2}{N_i \cdot \sum_{k \in S_i} a_k^2}.$$

Based on this, the generalized Jain's air-time fairness index is defined as

$$\overline{J} = \frac{1}{N} \cdot \sum_{k=1}^{N} J_k.$$

(Note that node i fairly shares air-time with its neighbor nodes if $J_i = 1$, and $\overline{J} = 1$ implies that $J_i = 1$ for all i.)

From these fairness indices, we can derive the following proposition. Consider a network where there is at least a node in the carrier sensing range of each node, and we will call this an interconnected network.

Proposition 4.1 (condition for max-min air-time fair). *The air-time $\overrightarrow{a}$ for an interconnected random access network is max-min air-time fair if and only if a_i, $i = 1, 2, \ldots, N$, are feasible and maximized under the constraint of $\overline{J} = 1$.*

Proof. (If part) Since $\overline{J} = 1$, each node has the same air-time as its neighbor nodes. This implies that $a_i = a_j$, for all $j = 1, 2, \ldots, N$. Let a' be the maximum air-time that a node can have under the constraint $\overline{J} = 1$. Then, $a_i = a'$ for all i. Assume that $\overrightarrow{a'} = (a', a', \ldots, a')^T$ is not max-min air-time fair. Then there must be another air-time vector $\overrightarrow{a'_\Delta} = (a' + \Delta, a' + \Delta, \ldots, a' + \Delta)$. However, this is impossible because of the definition of a'.

(Only if part) When each node in a network increases its air-time from zero until there is a node that cannot increase its air-time. Let $\tilde{a}$ be the air time value of each node at this time. In this case, if some of the nodes try to increase their air-time from $\tilde{a}$, there must be a node that needs to decrease its air-time from $\tilde{a}$, and consequently the minimum air-time value of a node in the network decreases from $\tilde{a}$. Therefore, $\tilde{a}$ is the max-min air-time value that each node can have, and consequently $\overline{J} = 1$. □

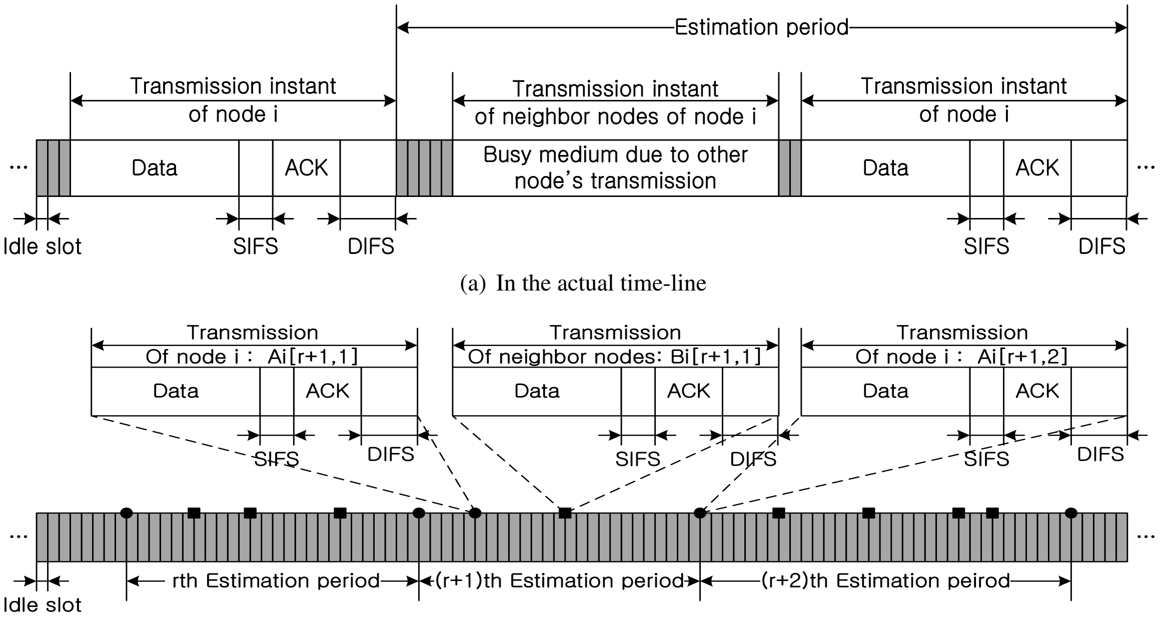
From Proposition 4.1, we can achieve max-min air-time fair if we can maximize a_i and a_j and maintaining the condition $a_i = a_j$, for all $j \in S_i$ and for each node i. This is an idealistic goal in a general ad-hoc network, but is a right direction to improve max-min air-time fairness in real circumstances. Therefore, we need a fairness measure for a node to announce in order to improve max-min air-time fairness. For this, each node estimates its busy time ratio, which is defined as the following.

Definition 4.5 (busy time ratio of node i)**.** *The busy time ratio of node i is the channel busy time owing to the transmission by node i with respect to the total channel busy time, which can be expressed as*

$$\alpha_i \;=\; \frac{a_i}{b_i}.$$

Here, $b_i = B_i/T$ and B_i is the time duration that node i senses the channel busy including its own transmission duration in time interval T.

Now, we present how each node estimates α_i using only measurable MAC layer statistics. We define the $estimation\ period$ for node i as a time period consisting of backoff slots (i.e., idle slots), busy medium times due to other nodes' transmission, node i's data transmission and the corresponding SIFS, ACK transmission, and DIFS as shown Fig. 4.1(a). (The busy medium time of node i is the time that node i senses its channel busy.) For ease of explanation, these terms, excluding the backoff slots and the busy medium time, are merged into a $transmission\ instant$. As each transmission starts and ends at the slot boundaries, we can abstractly draw the transmission instants and busy medium times due to other nodes' transmission as black dots, as shown in Fig. 4.1(b). An estimation period of node i starts from the end of its current


Estimation period
Transmission instant of node i
Transmission instant of neighbor nodes of node i
Transmission instant of node i
...
Data
ACK
Busy medium due to other node's transmission
Data
ACK
...
Idle slot
SIFS
DIFS
SIFS
DIFS
Transmission Of node i : Ai[r+1,1]
Transmission Of neighbor nodes: Bi[r+1,1]
Transmission Of node i : Ai[r+1,2]
Data
ACK
Data
ACK
Data
ACK
SIFS
DIFS
SIFS
DIFS
SIFS
DIFS
...
...
Idle slot
rth Estimation period
(r+1)th Estimation period
(r+2)th Estimation peirod

(a) In the actual time-line

(b) Abstraction

Figure 4.1 Estimation cycle of MAF-MAC for node i

transmission, and ends at the end of its next transmission if it can sense at least one transmission of its neighbor nodes between these transmissions. If not, the estimation period is extended to the end of its later transmission until at least one of its neighbor nodes transmits. It is noted that estimation period is not fixed, but this does not give any difficulty because we do not use the estimation period explicitly in the implementation of MAF-MAC. We denote the busy time duration owing to the kth transmission by node i in the rth estimation period as $A_i[r, k]$. Similarly, the busy time duration owing to the lth transmission by the neighbor nodes of node i in the rth estimation period is denoted as $B_i[r, l]$. And, $n_i[r]$ and $n_o[r]$ represent the number of transmissions by node i and the other neighbor nodes in the rth estimation cycle, respectively. Since there must be one or more transmissions of other nodes in an estimation period, $n_o[r]$ is greater than 0. Then, $a_i[r]$ and $b_i[r]$ can be expressed as the following

$$\begin{aligned} a_i[r] &= \frac{\sum_{k=1}^{n_i[r]} A_i[r,k]}{T_i[r]}, &(4.1)\\ b_i[r] &= \frac{\sum_{l=1}^{n_o[r]} B_i[r,l] + \sum_{k=1}^{n_i[r]} A_i[r,k]}{T_i[r]}, &(4.2)\end{aligned}$$

where $T_i[r]$ is the time duration of the rth estimation period for node i. Node i estimates its busy time ratio α_i after the rth estimation period as

$$\alpha_i[r] = \frac{a_i[r]}{b_i[r]} = \frac{\sum_{k=1}^{n_i[r]} A_i[r,k]}{\sum_{l=1}^{n_o[r]} B_i[r,l] + \sum_{k=1}^{n_i[r]} A_i[r,k]}.$$

Finally, α_i is low pass filtered with a coefficient ρ $(0 < \rho < 1)$ after the rth estimation period to produce $\widehat{\alpha_i}[r+1]$ as

$$\widehat{\alpha_i}[r+1] = \rho \cdot \widehat{\alpha_i}[r] + (1-\rho) \cdot \alpha_i[r]$$

Let $\overline{a_i}$ and $\overline{b_i}$ be the expect values of $a_i[r]$ and $b_i[r]$ in steady state, respectively. Then, for sufficiently large r, we can assume that $a_i[r] \approx \overline{a_i}$ and $b_i[r] \approx \overline{b_i}$, and $\widehat{\alpha_i}[r+1]$ can be approximated as

$$\widehat{\alpha_i}[r+1] \approx \frac{\overline{a_i}}{\overline{b_i}}$$

Let $q_{i,j}$ is the conditional probability that node j senses the channel idle given that node i senses the channel idle. If the neighbor nodes of nodes i and j are identical, i.e., $S_i = S_j$ the conditional probability $q_{i,j}$ then becomes 1. Assume that $q_{i,j} = 1, \forall j \in S_i$, (single-cell case) then the following proposition can be made.

Proposition 4.2 (a condition for approximate max-min air-time fair in a single-cell network)**.**

The air-time vector $\overrightarrow{a}$ for a wireless random access network is approximately max-min air-time fair if and only if $\widehat{\alpha_i}$ and $\widehat{\alpha_j}$ are maximized under the constraint of $\widehat{\alpha_i} = \widehat{\alpha_j}$ when $q_{i,j} = 1$, for all $j \in S_i$.

Proof. Note that $\overline{b_i} = \overline{b_j}$, for all $j \in S_i$, because node i and j sense the identical channel. Thus, we can see that $\overline{a_i} \approx \overline{a_j}$ from the $\frac{\overline{a_i}}{\overline{b_i}} \approx \widehat{\alpha_i} = \widehat{\alpha_j} \approx \frac{\overline{a_j}}{\overline{b_j}}$, and consequently $\overline{J} \approx 1$. Moreover, $\widehat{\alpha_i}$ and $\widehat{\alpha_j}$ are proportional to a_i and a_j, respectively. Therefore, a_i and a_j are maximized when $\widehat{\alpha_i}$ and $\widehat{\alpha_j}$ are maximized. From Proposition 4.1, the air-times of nodes in a wireless network is approximately max-min air-time fair if and only if $\widehat{\alpha_i}$ and $\widehat{\alpha_j}$ are maximized under the constraint of $\widehat{\alpha_i} = \widehat{\alpha_j}$ in a single-cell network. □

However, Proposition 4.2 is applicable only to single cell networks. Because the carrier sensing areas of nodes i and j are different in ad-hoc networks, and consequently $q_{i,j}$ is less than 1 in practice. Therefore, the following assumption is required to extend Proposition 4.2 for general ad-hoc networks.

Assumption 4.1. *When each node in an interconnected random access network greedily accesses the channel to maximize $\widehat{\alpha_i}$s while satisfying the constraint $\widehat{\alpha_i} = \widehat{\alpha_j}$, for all i and $j \in S_i$, the channel is busy for most of the time. Consequently, nodes i and j, for all i and $j \in S_i$, sense the channel busy for most of the time in steady state, and*

$$\overline{b_i} = \overline{b_j} \approx 1 \tag{4.3}$$

Then, we can derive the following proposition under the Assumption 4.1.

Proposition 4.3 (a condition for approximate max-min air-time fair). *Under the Assumption 4.1, air-time vector $\overrightarrow{a}$ for an interconnected random access network is approximately max-min air-time fair if and only if nodes i and j greedily access the channel to maximize $\widehat{\alpha_i}$ and $\widehat{\alpha_j}$ while satisfying the constraint of $\widehat{\alpha_i} = \widehat{\alpha_j}$, for all $j \in S_i$.*

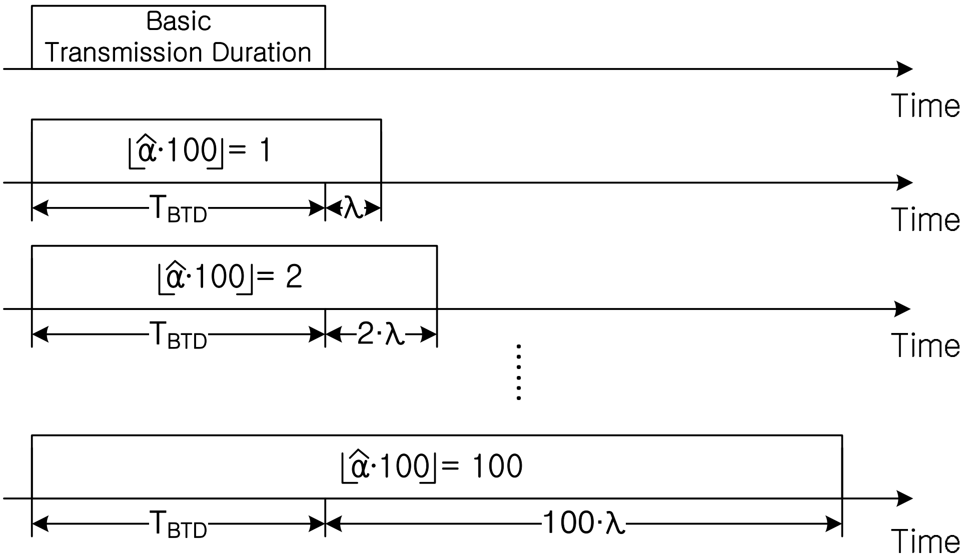


Figure 4.2 Busy time ratio in percent coding

Proof. Assume that nodes i and j greedily access the channel to maximize $\widehat{\alpha_i}$ and $\widehat{\alpha_j}$, and $\overline{b_i} = \overline{b_j}$ by Assumption 4.1. Then, the proof of this proposition follows similar to Proposition 4.2 because $a_i \approx a_j$ can be derived from $\widehat{\alpha_i} = \widehat{\alpha_j}$. □

In MAF-MAC, node i determines its transmission duration $T_{send,i}$ in accordance with $\widehat{\alpha_i}$ as

$$T_{send,i} \quad = \quad T_{BTD} + \lfloor \widehat{\alpha_i} * 100 \rfloor * \lambda \tag{4.4}$$

, where $\lfloor \cdot \rfloor$ is the notation for round down operation. Note that each frame size can be precisely adjusted and each transmission can last for a specified duration in MAF-MAC by using the modified fragmentation technique and zero padding, which were described in Chapter 2. That is, each node can announce its busy time ratio in percent ($\%$) by adjusting its transmission duration as can be seen in Fig. 4.2. Then, the other neighbor nodes of node i can deduce $\widehat{\alpha_i}$ from the transmission duration.

After node i deduces $\widehat{\alpha_j}$s of its neighbor nodes based on their transmission duration, node

i adjusts its attempt probability τ_i in an additive increase and multiplicative (AIMD) manner to maximize $\widehat{\alpha_i}$ and $\widehat{\alpha_j}$ while try to achieve the fairness condition $\widehat{\alpha_i} = \widehat{\alpha_j}$, for all $j \in S_i$, as the following.

$$\tau_i = \begin{cases} \mu \cdot \tau_i, \text{ if } \widehat{\alpha_i} > \widehat{\alpha_j} \\ \tau_i + c, \text{ otherwise,} \end{cases} \tag{4.5}$$

where μ $(0 < \mu < 1)$ and c (> 0) design parameters. It is noted that there is no need to identify the transmission node. If node i guesses that it uses more air-time than the other neighbor nodes (i.e., $\widehat{\alpha_i} > \widehat{\alpha_j}$), it multiplicatively decreases its attempt probability τ_i to yield channel resources to the other neighbor nodes. Otherwise, it additively increases τ_i to get its fair share. We can write (4.5) using the contention window CW_i of node i because $\tau_i = 2/(CW_i + 1)$ [59]. Also we make sure that CW_i is in the range of [CW_{min},CW_{max}].

$$CW_i = \begin{cases} \min\{\frac{1}{\mu} \cdot (CW_i + 1) - 1, CW_{max}\}, \text{ if } \widehat{\alpha_i} > \widehat{\alpha_j}, \\ \max\{\dfrac{2 \cdot (CW_i + 1)}{2 + c \cdot (CW_i + 1)} - 1, CW_{min}\}, \text{ otherwise.} \end{cases} \tag{4.6}$$

In this way, MAF-MAC tries to improve max-min air-time fairness in distributed manner by using the transmission duration.

4.3 MAF-MAC in various circumstances

In this section, we study several typical cases to see how MAF-MAC works to improve max-min air-time fairness in ad-hoc networks.

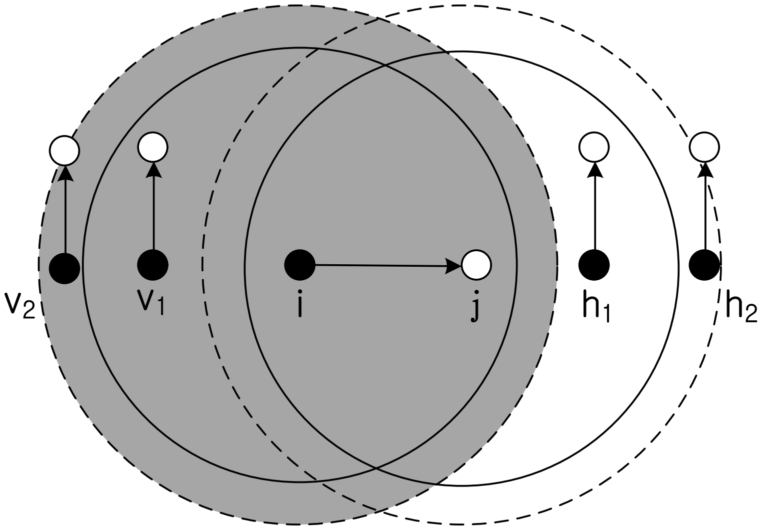


Figure 4.3 Announcing range of node i in MAF-MAC

4.3.1 Handling the Hidden Node Problem

In IEEE 802.11 ad-hoc networks, we need to take the hidden node problem into consideration, because there may exist hidden nodes in a network, which can results in significant fairness degradation [6, 7]. Let the announcing range of node i be the range that node i can implicitly inform its $\widehat{\alpha_i}$ to other neighbor nodes by its transmission duration, as can be seen in Fig. 4.3. The gray area in Fig. 4.3 represents the announcing range of node i when node i transmits a data frame to node j, while the circles with the solid-line and the dotted-line indicate the transmission range and carrier sensing range, respectively. The announcing range and the carrier sensing range are the same in this case. Then, nodes v_1 and v_2, which are in the announcing range of node i, can sense the transmissions of node i, and consequently deduce $\widehat{\alpha_i}$ from the transmission duration of node i. However, nodes h_1 and h_2, which are hidden nodes to node i, cannot senses the transmissions of node i, and thus cannot adjust their CWs in accordance with (4.6) because they do not know $\widehat{\alpha_i}$. Moreover, the hidden nodes interfere the transmissions of node i, which can severely degrade max-min air-time fairness. To improve max-min air-time fairness even when there are hidden nodes in ad-hoc networks, MAF-MAC adopts the the hidden node detection mechanism [8], extension of the effective CTS range and Receiver-Oriented Contention (ROC) [9], which are summarized in Appendices A and B, respectively. In the following, we explain how these schemes can be used in MAF-MAC.

Hidden node detection mechanism

Because MAF-MAC also uses the fragment aggregation and block ACK features, the hidden node detection mechanism in [8] can be easily adopted to MAF-MAC except for the calculation of THR_i^B and THR_i^R, where THR_i^B and THR_i^R are the estimated throughputs without/with using the RTS/CTS control frame exchange, respectively. Note that a fragment is the basic unit used in retransmission rather than a packet in MAF-MAC. Therefore, the throughput (THR) is calculated as

$$THR = \frac{n_{s,frag} \cdot S_{frag}}{T_{data} + T_{oh}}, \tag{4.7}$$

where $n_{s,frag}$ and S_{frag} are the number of successfully transmitted fragments and the fragment size, respectively. The number of successfully transmitted fragments can be estimated as

$$n_{s,frag} = \begin{cases} (1 - p_{i,E}^{hid})(1 - p_{i,P}^{hid})n_{tf}, & \text{for basic access mode,} \\ n_{tf}, & \text{for RTS/CTS mode,} \end{cases} \tag{4.8}$$

where n_{tf} is the total number of fragments in a data frame. In MAF-MAC, node i calculates THR_i^B and THR_i^R based on $p_{i,E}^{hid}$ and $p_{i,P}^{hid}$, which are estimated as in [8] (see Appendix A), and employs the RTS/CTS exchange when THR_i^R is larger than THR_i^B. In the simulation of section 4.4, the parameters $\eta^{(E2)}$ and $\eta^{(P2)}$ for MAF-MAC are set to 0.1 and 0.05 respectively, as in [8].

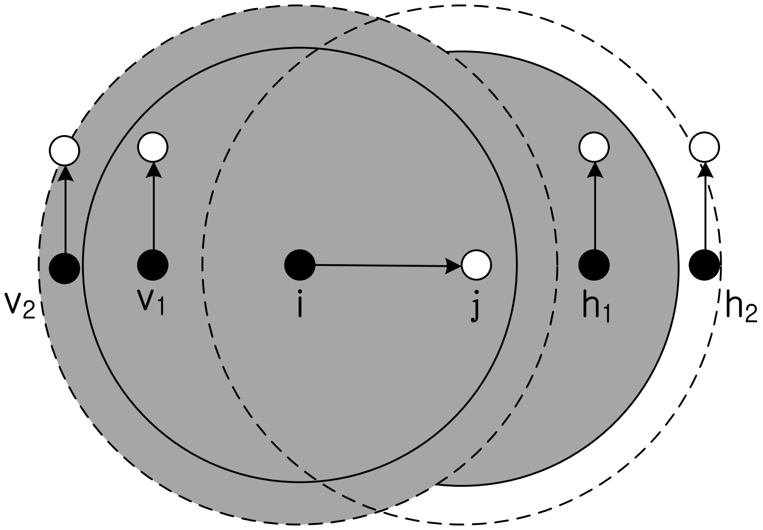


Figure 4.4 Announcing range of node i in MAF-MAC when a CTS frame is used

Note that there is a duration field in a CTS frame. If a node can decode the CTS frame, it can deduce $\widehat{\alpha}$ of the upcoming transmission by using the duration field in a CTS frame. Therefore, the announcing range of node i can be extended when the RTS/CTS mechanism is used, as can be seen in Fig. 4.4. When node j transmits a CTS frame, node h_1 in Fig. 4.4 can decode the CTS frame, and thus it can deduce $\widehat{\alpha_i}$ and properly adjust its CW value according to (4.6).

Extending the effective CTS range

There are two major problems to be resolved in extending the effective CTS range (see Appendix B), which are how to identify a CTS frame when a node can only sense the transmission of a data frame, and how to set the NAV value to protect the upcoming data transmission. Since MAF-MAC uses the block ACK feature, a CTS frame can be differentiated from an ACK frame by its transmission duration as in [9]. After identifying a CTS frame, each node has to defer channel access to protect the upcoming data transmission. In [9], a node simply sets NAV to T_{ref} after identifying the CTS frame, because the average transmission time of a data frame T_{data} is close to T_{ref} by using the Frame size Adaptation (FA) scheme.

In MAF-MAC, each node has different transmission duration according to $\widehat{\alpha} \cdot 100$, which is always less than 100. Therefore, the upcoming data transmission is always protected if the NAV value is set to the maximum transmission duration $T_{send}^{max} = BTD + 100 \cdot \lambda$ after identifying a CTS frame. However, a large portion of channel resource may be wasted when the $\widehat{\alpha}$ value of transmitting node is small. Note that the hidden node problem can occur only when there are three or more flows in an ad-hoc network. Therefore, we use CTS frames only when there are at least three flows in a network. In this case, $\widehat{\alpha} \cdot 100$ should be less than or equal to 33% in order to achieve max-min air-time fairness, and consequently the average transmission duration of a node is less than $T_{BTD} + 34 \cdot \lambda$. Thus, each node set its NAV value to $T_{BTD} + 34 \cdot \lambda$ after identifying a CTS frame to reduce wasted air-time and to protect upcoming data transmission. In this way, even though node h_2 in Fig. 4.4 does not know $\widehat{\alpha_i}$, it can avoid interfering the transmission of node i when it senses a CTS frame from node j by setting its NAV value appropriately.

Receiver-oriented contention (ROC) mechanism

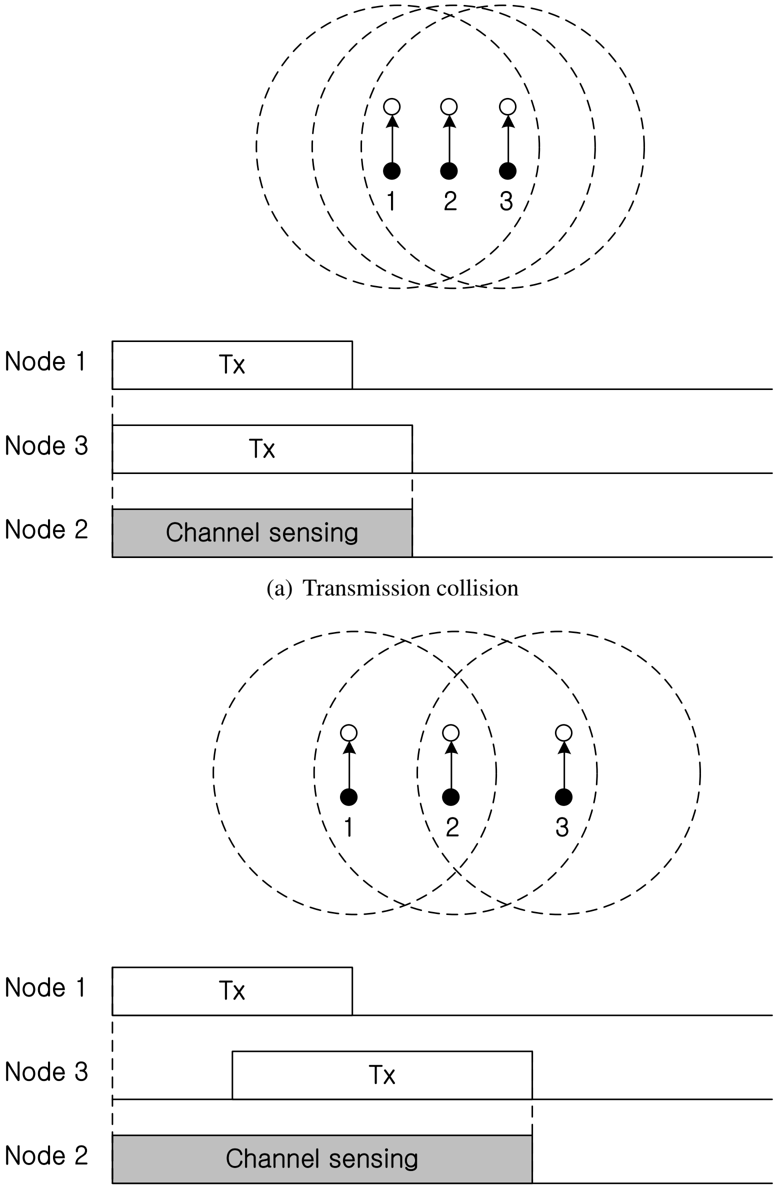
To more effectively resolve the hidden node problem, the Receiver-Oriented Contention (ROC) mechanism can be adopted to MAF-MAC when hidden nodes are detected. However, there are some issues to solve before adopting the ROC mechanism. When node i transmits a data frame to node j, and these nodes decide to use the ROC mechanism because of hidden nodes. Then, node j initiates a transmission when it ends the backoff procedure by transmitting a CTS frame to node i. However, node j does not know the exact value of $\widehat{\alpha_i}$, which is important information for the nodes that cannot sense the transmission of node i but can decode the CTS frame from node j. Hence, node j set the time duration in the CTS duration field to the previous $\widehat{\alpha_i}$, which was obtained from the previous transmission of node i. Because of hidden nodes and the difference in the carrier sensing ranges of nodes i and j, node j may not know most of values of $\widehat{\alpha_k}$, for all $k \in S_i$. Therefore, node j cannot adjust its CW value in accordance with (4.6), and contends based on the Binary Exponential Backoff (BEB) mechanism as in the IEEE 802.11 DCF. For this reason, even though the hidden node problem can effectively be resolved, the fairness performance can degrade when the ROC mechanism is adopted to MAF-MAC.

4.3.2 Handling Transmission Collision and Overlap

In MAF-MAC, each node deduces $\widehat{\alpha}$ of current transmission based on transmission duration, and adjusts its CW value according to $\widehat{\alpha}$. A transmission collision occurs when two or more nodes, which are in the carrier sensing range of each other, transmit a data frame at the same slot time. On the other hand, a transmission overlap occurs when transmissions of two or more nodes that are not in the carrier sensing range of each other are overlapped. Figure 4.5 shows the difference between the transmission collision and overlap when nodes 1 and 3 transmits data. Note that nodes 1 and 3 are in the carrier sensing range of each other in Fig. 4.5(a), but not in the carrier sensing range in Fig. 4.5(b).

The transmission collision can be easily handled, because a node can deduce the largest $\widehat{\alpha}$ from the transmission that have the longest duration among the failed transmissions due to collision. Figure 4.5(a) shows that node 2 can deduce $\widehat{\alpha_3}$, where the transmission duration of node 3 is longer than that of node 1. Therefore, node 2 can adjust its CW_2 in accordance with (4.6) based on $\widehat{\alpha_3}$.

When two or more transmissions overlap, it is difficult for a node to correctly deduce the proper value of $\widehat{\alpha}$ based on the transmission duration as we can see in Fig. 4.5(b). Node 2 cannot know the values of $\widehat{\alpha_1}$ and $\widehat{\alpha_3}$ when the transmissions of node 1 and 3 overlap. If the overlapped transmission duration is longer than $BTD + 100 \cdot \lambda$, each node can deduce that the transmissions are overlapped and does not adjust its CW value. (In this example, $\widehat{\alpha_j}$ depends on the overlapping transmission duration of nodes 1 and 3, and j is neither 1 nor 3.) This



(a) Transmission collision

(b) Transmission overlap

Figure 4.5 Transmission collision and overlap

corresponds to $\widehat{\alpha_j} > 100$ in (4.6), and node i (= 2 in this example) does not change its CW_i. Otherwise, it cannot properly adjust its CW value, because it cannot correctly deduce $\widehat{\alpha}$ from the transmission duration. In general, a node that experiences transmission overlap is a starving node, and the nodes that uses more channel resource can deduce the $\widehat{\alpha}$ value of starving nodes from their transmission duration. That is, nodes 1 and 3 in Fig. 4.5(b) can correctly deduce $\widehat{\alpha_2}$ most of the time and can properly adjust their CWs when node 2, which is a starving node, transmits a data frame. Note that if node 2 starves and nodes 1 and 3 heavily use channel resource, the $\widehat{\alpha_2}$ value is smaller than $\widehat{\alpha_1}$ and $\widehat{\alpha_3}$. Therefore, nodes 1 and 3 can see that node 2 is starving from the $\widehat{\alpha_2}$ value and yield their transmission opportunities to node 2 according to (4.6). This in turn reduces the transmission overlaps and node 2 can correctly deduce $\widehat{\alpha_1}$ and $\widehat{\alpha_3}$ more frequently. In this way, MAF-MAC can handle the starvation problem and improve max-min air-time fairness even when the transmission overlap occurs in ad-hoc networks.

4.3.3 Handling Unsaturated Nodes

An unsaturated node, whose aggregated length of packets in the queue is smaller than the required frame body length (S_{fb}), cannot correctly inform other nodes of its $\widehat{\alpha}$ using the transmission duration. Let T_{agg}^{us} be the transmission duration for an unsaturated node transmitting aggregated packets in its queue. In MAF-MAC, each node does not adjust its CW value in accordance with (4.6) if the current transmission duration is shorter than T_{BTD}. Therefore, when T_{agg}^{us} is shorter than T_{BTD}, which corresponds to $\widehat{\alpha_j} < 0$, a node simply transmits a data frame when its backoff counter becomes zero, because it does not affect the CW values of the other nodes. Then, (4.6) is modified as

$$CW_i = \begin{cases} \min\{\frac{1}{\mu} \cdot (CW_i + 1) - 1, CW_{max}\} & \text{, if } \widehat{\alpha_i} > \widehat{\alpha_j} \geq 0, \\ \max\{\dfrac{2 \cdot (CW_i + 1)}{2 + c \cdot (CW_i + 1)} - 1, CW_{min}\} & \text{, if } \widehat{\alpha_i} < \widehat{\alpha_j} \leq 100, \\ CW_i & \text{, if } \widehat{\alpha_j} < 0 \text{ or } \widehat{\alpha_j} > 100. \end{cases} \tag{4.9}$$

If the T_{agg}^{us} is longer than T_{BTD} but shorter than $T_{send} = T_{BTD} + \lfloor \widehat{\alpha} * 100 \rfloor * \lambda$, there are two options, i.e., waiting for the packets from the upper layer or using the zero-padding method to fill the data frame with zero bits if necessary. If $(T_{send} - T_{agg}^{us}) > \gamma$, where γ (> 0) is a design parameter, an unsaturated node waits packets from the upper layer until the aggregated packet length in the queue is greater than or equal to S_{fb}. Otherwise, it zero-pads the remaining data frame and transmits it as soon as the backoff procedure ends. There is tradeoff between transmission delay and MAC efficiency depending on the value of γ. If γ is zero, an unsaturated node always waits for the packets from the upper layer, and thus delay increases. As γ increases, it uses the zero-padding more frequently, and thus delay may not in decrease, but the MAC efficiency degrades. (In the simulation of Section 4.4, we set γ to $50 \cdot \lambda$.)

4.3.4 Enhancing Channel Utilization

Although the max-min air-time fairness is improved as $\overline{J}$ approaches 1, the channel can be underutilized when the nodes in a network are not uniformly distributed as in Fig. 4.6. Assume that m nodes are placed in the right-side of carrier sensing range of node 2. Then, the maximum air-time value that each node can have under the constraint $\overline{J} = 1$ is bounded by $\frac{1}{m+2}$. Therefore, the gray area in Fig. 4.6 may be underutilized as m increases, because the sum of air-time of nodes 1 and 2 is less than $\frac{2}{m+2}$. In this case, node 1 notices that its channel is underutilized.

To enhance the channel utilization in such a non-uniformly distributed network even though

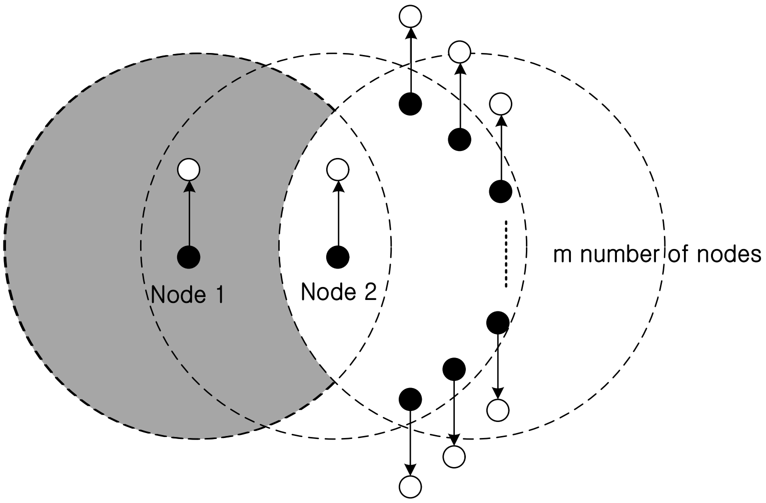


Figure 4.6 Channel can be underutilized when the nodes in a network are not uniformly distributed

it may decrease the air-time fairness a little, node i estimates $b_i[r]$ after the rth estimation period as in (4.1), and $b_i[r]$ is low pass filtered as the following.

$$\widehat{b_i}[r+1] \quad = \quad \rho \cdot \widehat{b_i}[r] + (1-\rho) \cdot b_i[r],$$

where $0 < \rho < 1$. If $\widehat{b_i}[r+1]$ is less than ζ ($0 < \zeta < 1$), node i deduces that the channel is underutilized, and simply decreases CW_i (i.e., increases attempt probability) regardless of value of $\widehat{\alpha_j}$ whenever it senses the transmission of its neighbor nodes. Therefore, (4.9) is finally modified as the following.

$$CW_i \quad = \begin{cases} \dfrac{2 \cdot (CW_i + 1)}{2 + c \cdot (CW_i + 1)} - 1 & \text{, if } \widehat{b_i} < \zeta, \\ \text{Adjust } CW_i \text{ according to (4.9)} & \text{, otherwise.} \end{cases} \tag{4.10}$$

4.4 Performance evaluation

In this section, we evaluate MAF-MAC via simulation using the ns-2 simulator [60], and compare it with the following five schemes.

- **BASIC** : This is a baseline scheme that only employs DCF.
- **BASIC(HD/E)** : This scheme employs the RTS/CTS mechanism with the extended CTS range when hidden nodes are detected, which was studied in [8].
- **BASIC(ROC/HD/E)** : This scheme employs the ROC mechanism with the extended CTS range when hidden nodes are detected, which was studied in [9].
- **MAF-MAC** : This scheme employs MAF-MAC instead of DCF without any hidden node detection/resolution mechanisms.
- **MAF(HD/E)** : This scheme employs MAF-MAC instead of DCF, together with the RTS/CTS mechanism and the extended CTS range when hidden nodes are detected.
- **MAF(ROC/HD/E)** : This scheme employs MAF-MAC, together with ROC mechanism and the extended CTS range when hidden nodes are detected.

We set λ to 4μs, because it is the minimum time required for packet detection/clear channel assessment (CCA) in 802.11n [11,12]. The values of ρ, μ and c in (4.9) for MAF-MAC were set to 0.8, $1/1.2$ and 0.002, respectively, and the basic rate was set to 6.5Mbps. We set path loss and data rate of each link according to distance between sender and receiver of the link. (Note that the data rate of a link is 65Mbps if the data rate is not explicitly mentioned, and the transmission and carrier sensing range were set to 250m and 400m, respectively.) The frame size in BASIC was always set to 10240byte, and and T_{BTD} in the schemes based on MAF-MAC was 1.3 ms, which corresponds to the transmission time of a 10240byte frame at 65Mbps. The average data transmission duration in BASIC(HD/E) and BASIC(ROC/HD/E) was also set to 1.3ms by applying the frame size adaptation scheme that tried to improve air-time fairness in multi-rate environment [9]. The size of a packet in both BASIC and MAF-MAC was 2048byte whereas a fragment in MAF-MAC was 256byte long. Corrupted data were retransmitted in packets for the schemes based on BASIC and in fragments for the schemes based on MAF-MAC. The CW_{max} and CW_{min} were always set to 32 and 1024, respectively. Each node always had packets to transmit, except for the simulation in the case where there was an unsaturated node in an ad-hoc network.

Since MAF-MAC adjusts the CW values to make neighbor nodes have the same busy time ratio, which in turn tries to achieve max-min air-time fairness, not throughput, we need some measures that can show the degree of channel utilization by a node in ad-hoc network. We define the $individual\ channel\ occupation\ ratio$ of link k ($C_{oc,k}$) and $avaerage\ channel\ occupation\ ratio$ (C_{oc}) as

$$\begin{aligned} C_{oc,k} &= \begin{cases} \dfrac{T^s_{busy,k}}{T^s_{busy,k}+T^s_{idle,k}}, & \text{if } k \in S_{SI} \\ \dfrac{T^r_{busy,k}}{T^r_{busy,k}+T^r_{idle,k}}, & \text{if } k \in S_{RI} \end{cases} \\ C_{oc} &= \frac{1}{N}\cdot\sum_{k=1}^{N} C_{oc,k}. \end{aligned}$$

Here, S_{RI} and S_{SI} are the sets of links, in which a sender or a receiver initiates a transmission, respectively. Also, $T^s_{busy,k}$ ($T^r_{busy,k}$) and $T^s_{idle,k}$ ($T^r_{idle,k}$) represent the time duration that a sender (receiver) of link k judges that the channel is busy and idle, respectively. We estimate C_{oc} to see whether the channel is busy or not from the view point of the transmission initiator. When C_{oc} is close to 1, we can say that the channel is fully used, and thus Assumption 1 is reasonable.

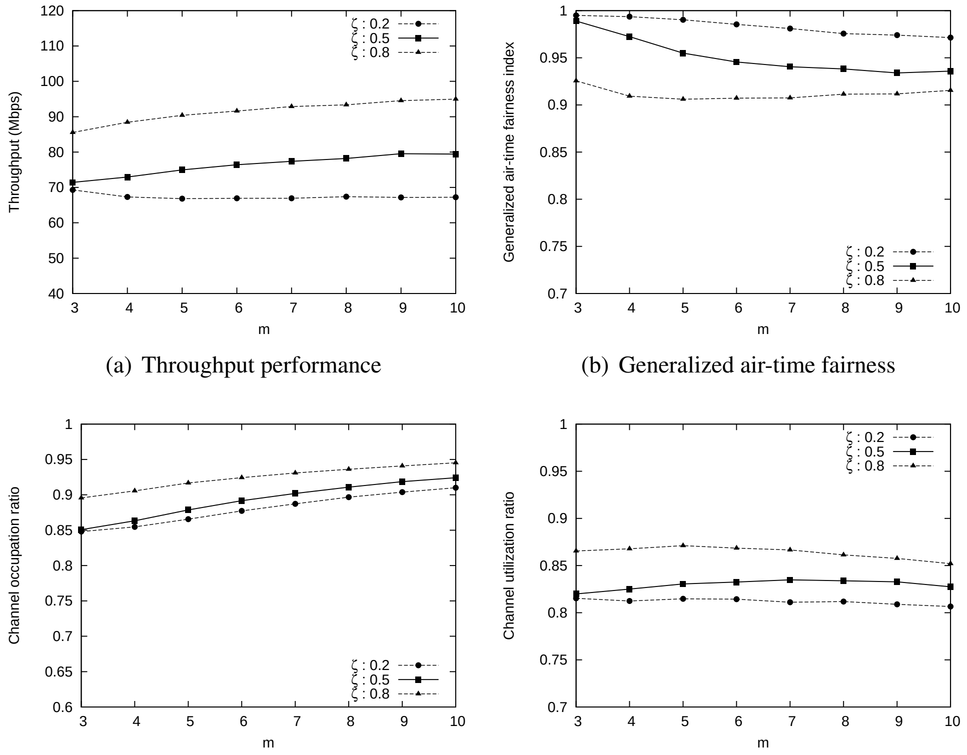
We also define the *individual channel utilization ratio* of link k ($C_{ut,k}$) and *average channel utilization ratio* (C_{ut}) as

$$\begin{aligned} C_{ut,k} &= C_{oc,k} \cdot R_{ta}^{sa} \cdot p_{s,k}^{frag}, \\ C_{ut} &= \frac{1}{N} \cdot \sum_{k=1}^{N} C_{ut,k}. \end{aligned}$$

Here, R_{ta}^{sa} is the ratio of aggregate air-time used for successful transmissions of link k with respect to total air-time of link k, and $p_{s,k}^{frag}$ is the probability that successfully transmitted fragments in link k are not corrupted. ($p_{s,k}^{frag}$ is always 1 when there is no channel noise.) Then, we can see that each node effectively utilizes its channel when C_{ut} is close to 1. Note that C_{ut} is an average that indicates how each node effectively uses the channel in its vicinity. The throughput is proportional to C_{ut} for a single-cell network. We use throughput, air-time fairness, C_{oc}, C_{ut} and average packet delay T_{del} as performance measures in the evaluation of network performance (T_{del} is the average amount of wait time before successfully transmitting a packet).

4.4.1 Effect of ζ on the performance of MAF-MAC

First, we study the effect of ζ on the performance of MAF-MAC for various value of m in Fig. 4.6. Figure 4.7 show the network performance corresponding to the value of ζ (= 0.2, 0.5 and 0.8). As ζ increases, node 1 in Fig. 4.6 more greedily accesses the channel, and thus the aggregate throughput, C_{oc} and C_{ut} are improved. However, the air-time fairness degrades a little, because node 2 can access the channel less frequently. In the following simulations, we set ζ to 0.5 to improve throughput performance while sacrificing air-time fairness performance as little as possible.



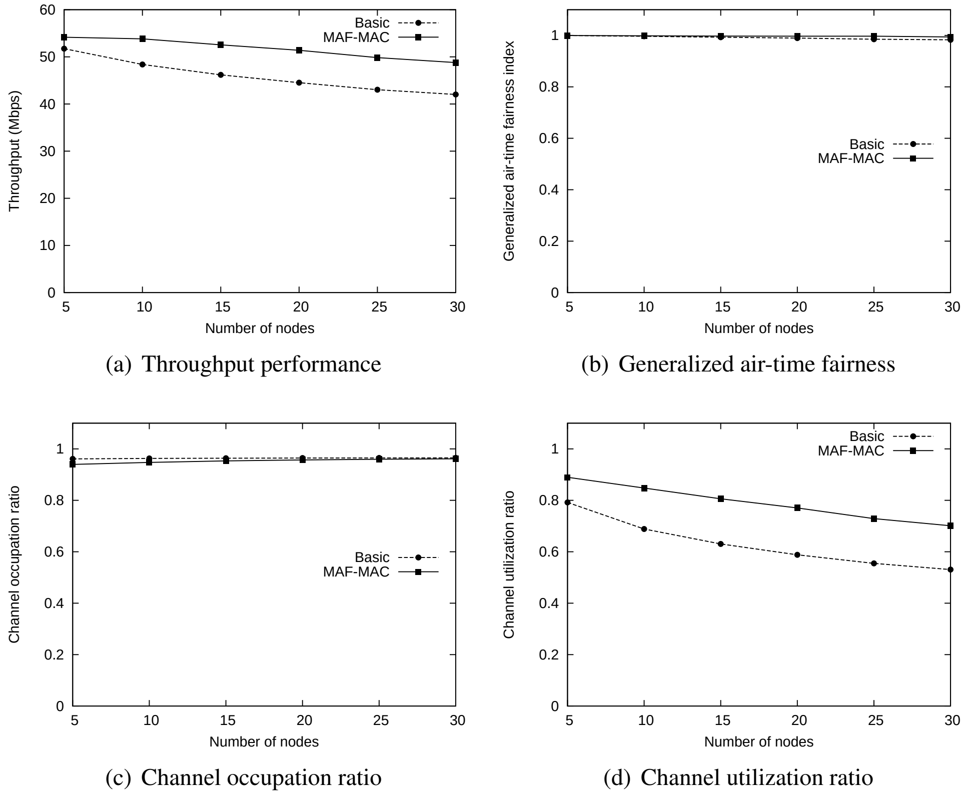
(a) Throughput performance

(b) Generalized air-time fairness

(c) Channel occupation ratio

(d) Channel utilization ratio

Figure 4.7 Effect of ζ on the network performance for various value of m



(a) Throughput performance

(b) Generalized air-time fairness

(c) Channel occupation ratio

(d) Channel utilization ratio

Figure 4.8 Network performance for various number of nodes in a single cell network

4.4.2 A single-cell network

We investigate the aggregate throughput, generalized air-time fairness, C_{oc} and C_{ut} when all the nodes are located in the carrier sensing range of each other, i.e., a single-cell network. In a single cell network, there is no hidden nodes in the network, and consequently, no difference in network performance between BASIC and BASIC with the hidden node detection/resloution schemes, and also between MAF-MAC and MAF-MAC with the hidden node detection/resolution schemes. Thus, we compare the network performances of BASIC and MAF-MAC only. Figure 4.8(c) shows that channel is nearly fully occupied regardless of the number of nodes whereas Fig. 4.8(d) shows that probability of successful data transmission de-

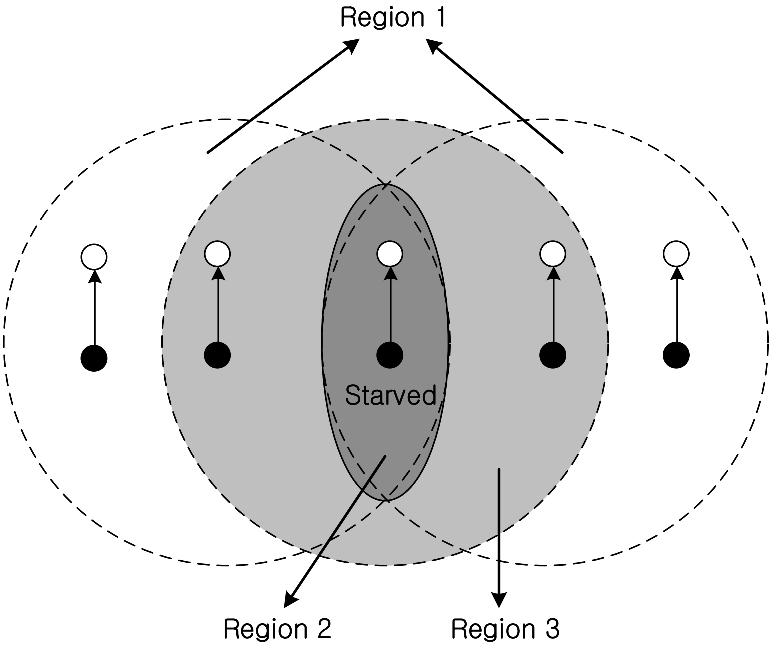


Figure 4.9 A simple starvation scenario without hidden nodes

creases as the the number of nodes increases resulting in the throughput decrease in Fig. 4.8(a). Figure 4.8 shows that MAF-MAC achieves near perfect air-time fairness and at the same time fully and effectively utilizes the channel, i.e., improves max-min air-time fairness regardless of the number of nodes. Moreover, MAF-MAC shows good throughput performance compared to BASIC as can be seen in Fig. 4.8(a). Although we have studied the delay performance of BASIC and MAF-MAC in a single-cell network, the simulation results are not presented because they were nearly the same in both schemes.

4.4.3 A simple scenario of the starvation problem

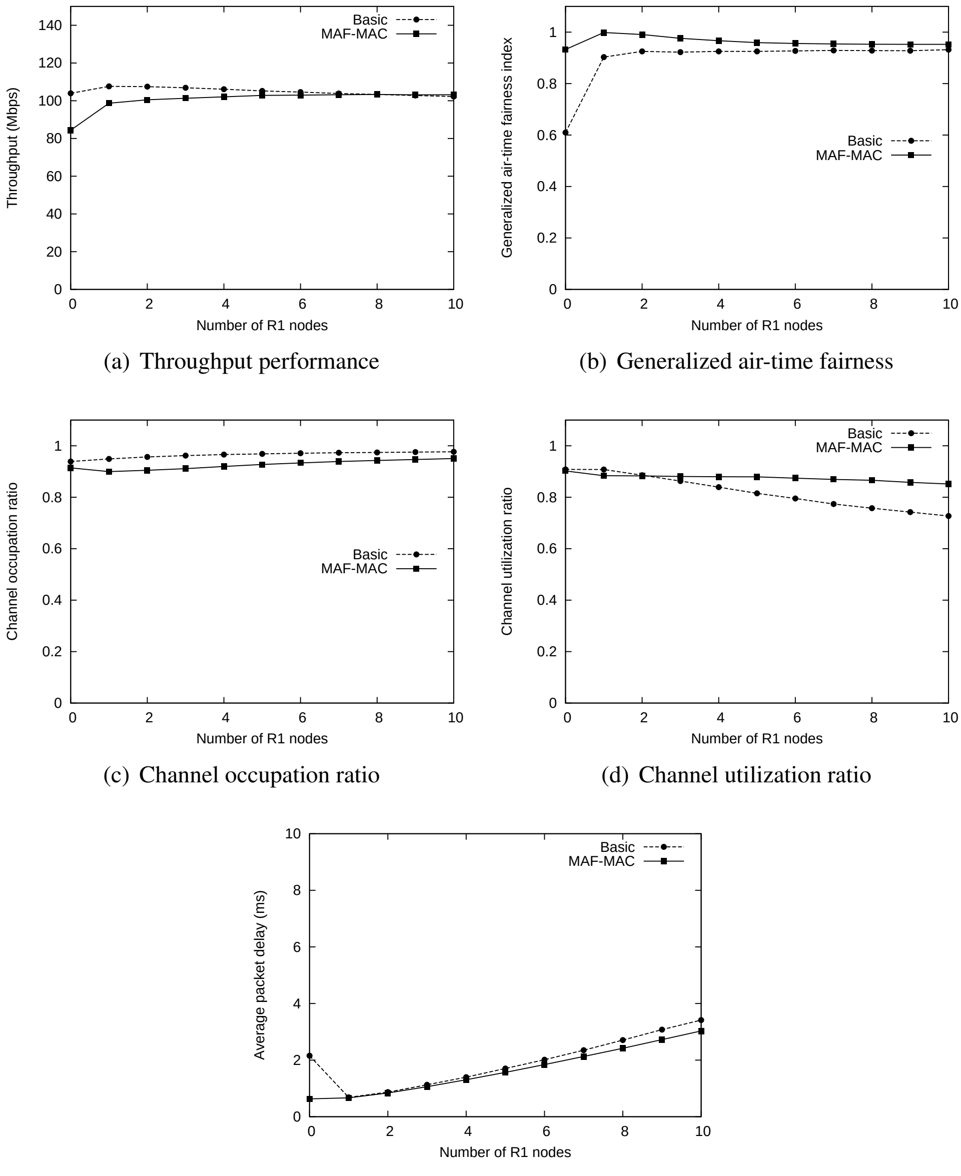
In this section, we investigate the network performance in a simple starvation scenario as in Fig. 4.9 to show that MAF-MAC effectively resolves the starvation problem. In this topology, the nodes in Region 2 starve because of the nodes in Region 3 that are in the carrier sensing range of the nodes in Region 2 but are not in the carrier sensing range of each other. To keep the nodes in Region 2 in starving state, we need two or more nodes that are out of the carrier sensing range of each other, in Region 3. Note that although the nodes in Region 1 are out of the carrier sensing range of the nodes in Region 2, they do not interfere the transmissions in Region 2, i.e., there are no hidden nodes. Therefore, we only compare the network performance of BASIC and MAF-MAC for various number of nodes in Regions 1 (R1), 2 (R2), and 3 (R3).

Network performance for various numbers of nodes in R1

We investigate the network performance for various numbers of R1 nodes when there are one and two nodes in R2 and R3, respectively. Note that R1 nodes can transmit data without interfering the transmission of R2 nodes, i.e., the channel can be spatially reused. Thus, the aggregate throughput can be greater than the data transmission rate (65 Mbps) as can be seen in Fig. 4.10(a). Figures 4.10(b) and 4.10(d) show that MAF-MAC maintains similar or better air-time fairness and higher C_{ut} compared to BASIC regardless of the number of R1 nodes. On the other hand, MAF-MAC and BASIC show nearly the same performance in the channel occupation ratio C_{oc} and packet delay T_{del}, as can be seen in Fig. 4.10(c) and 4.10(e).

Network performance for various numbers of nodes in R2

We evaluate the network performance for various numbers of R2 nodes when the numbers of R1 and R3 nodes are zero and two, respectively. Figure 4.11(b) shows that R2 nodes starved because of R3 nodes regardless of the number of R2 nodes, and thus the air-time fairness was severely degraded in the BASIC scheme. On the contrary, MAF-MAC successfully resolved the starvation problem and gave good air-time fairness performance for various numbers of R2 nodes. Because R2 nodes in BASIC had to wait for a long time to transmit a data frame, the packet delay of BASIC was much longer than that of MAF-MAC as shown in Fig. 4.11(e). Although Fig. 4.11(a) shows that the aggregate throughput is higher in the BASIC scheme



(a) Throughput performance

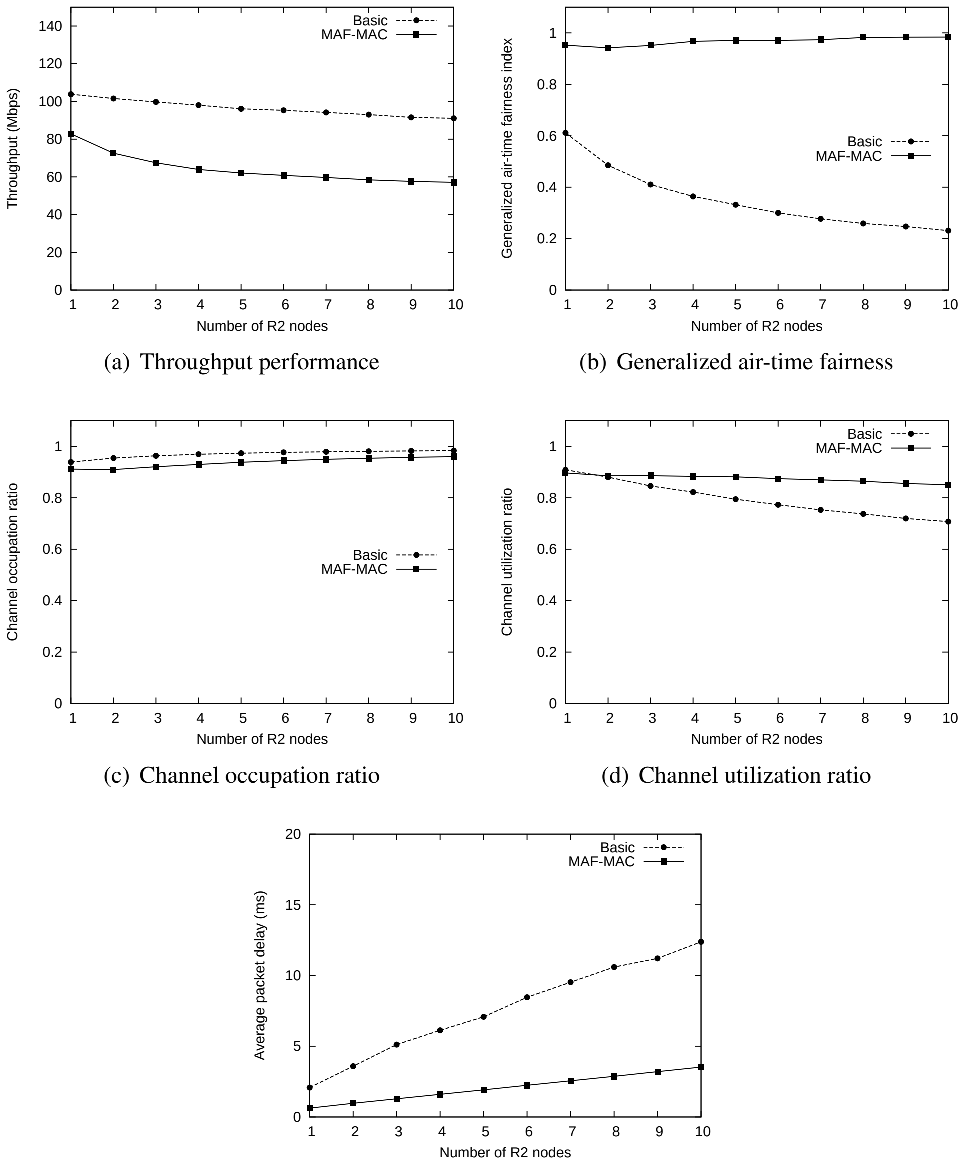
(b) Generalized air-time fairness

(c) Channel occupation ratio

(d) Channel utilization ratio

(e) Average packet delay

Figure 4.10 Network performance for various numbers of R1 nodes in Fig. 4.9



(a) Throughput performance

(b) Generalized air-time fairness

(c) Channel occupation ratio

(d) Channel utilization ratio

(e) Average packet delay

Figure 4.11 Network performance for various numbers of R2 nodes in Fig. 4.9

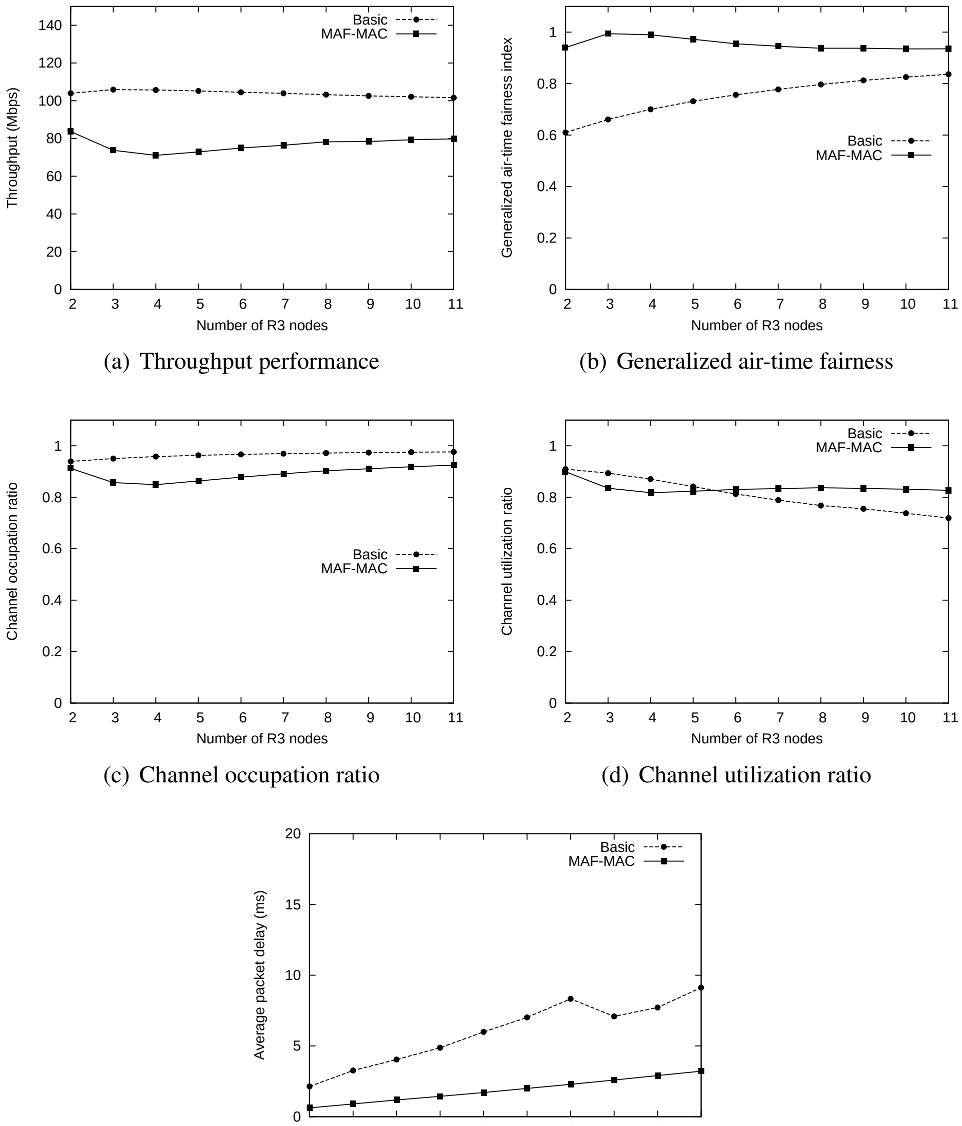
compared to MAF-MAC, we can see that C_{oc} of BASIC is nearly equal to that of MAF-MAC in Fig. 4.11(c). Furthermore, C_{ut} is lower in BASIC compared to MAF-MAC as can be seen in Fig. 4.11(d). This means that MAF-MAC fully and effectively utilized the channel resources while maintaining much higher air-time fairness compared to BASIC.

Network performance for various numbers of nodes in R3

Figure 4.12 shows the network performance for various numbers of R3 nodes when the numbers of R1 and R2 nodes are zero and one, respectively. The BASIC scheme shows better air-time fairness performance compared to the previous simulation results, but it still shows rather poor air-time fairness as can be seen in Fig. 4.12(b). MAF-MAC achieves better air-time fairness and delay performance compared to BASIC regardless of the number of R3 nodes as can be seen in Fig. 4.12(b) and 4.12(e). At the same time, Fig. 4.12(c) and 4.12(d) show that each node in MAF-MAC fully and effectively utilizes the channel resources. The simulation study shows that MAF-MAC achieves good air-time fairness and low packet delay with high C_{oc} and C_{ut} regardless of the number of nodes when there is no hidden node.

Simple starvation scenario with an unsaturated node

Figure 4.13 shows network performance when the numbers of nodes in R1, R2 and R3 are zero, six and two, respectively. All the nodes in R2 except one are saturated. In Fig. 4.13(a), Thr_T, Thr_U, and Thr_S represent the total throughput, aggregate throughputs of the unsaturated node and saturated nodes, respectively. The offered load of the unsaturated node varied every 5 s as $(1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1)$ Mbps. In MAF-MAC, the unsaturated node used a sufficient amount of channel bandwidth to transmit its offered load, and the saturated nodes shared the remaining bandwidth unused by the unsaturated node. However, the unsaturated node in the BASIC mode could not use its fair share, because nodes in R3 monopolized the channel



(a) Throughput performance

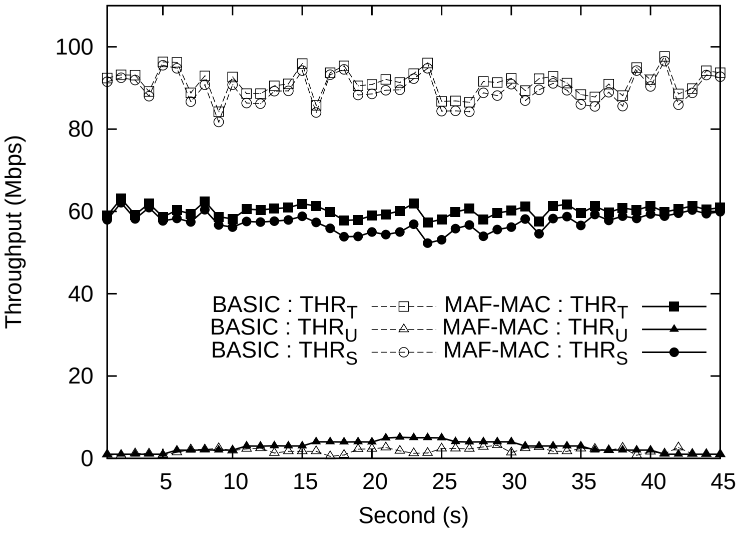
(b) Generalized air-time fairness

(c) Channel occupation ratio

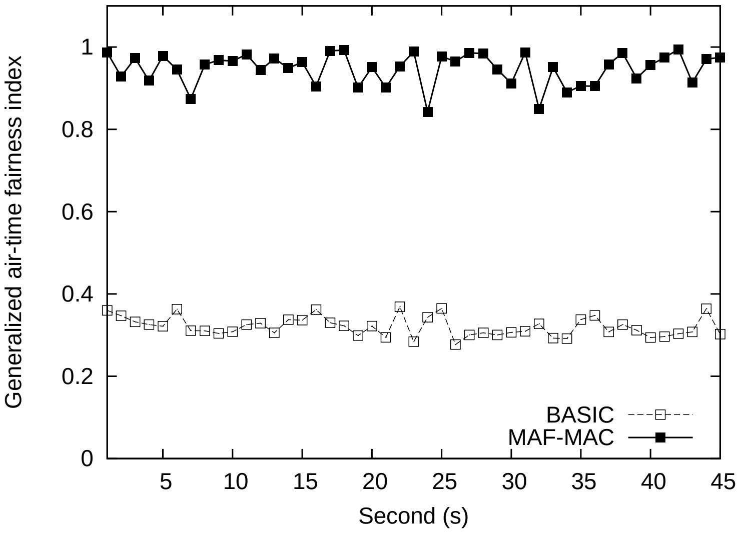
(d) Channel utilization ratio

(e) Average packet delay

Figure 4.12 Network performance for various numbers of R3 nodes in Fig. 4.9



(a) Throughput performance



(b) Generalized air-time fairness except an unsaturated node

Figure 4.13 Network performance when there is an unsaturated node in region 2

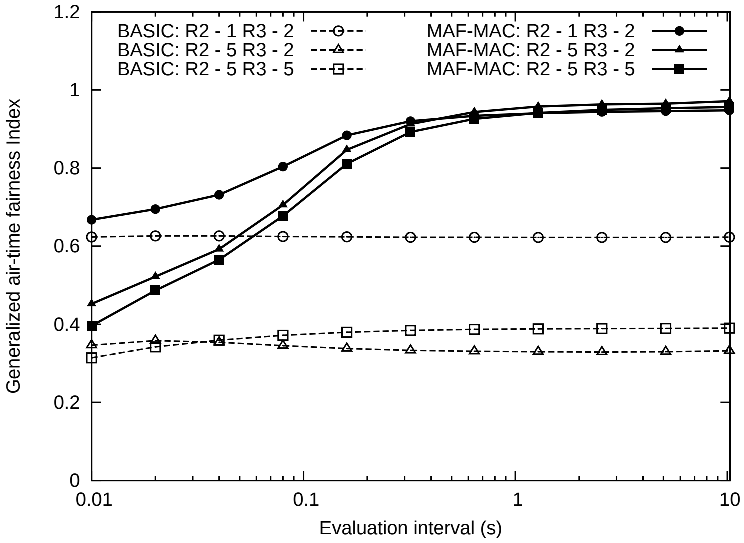
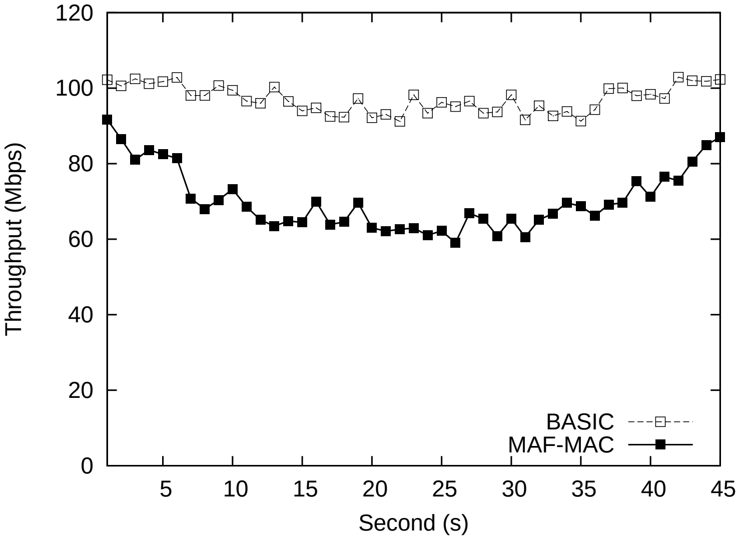


Figure 4.14 Short-term generalized air-time fairness for various number of R2 and R3 nodes

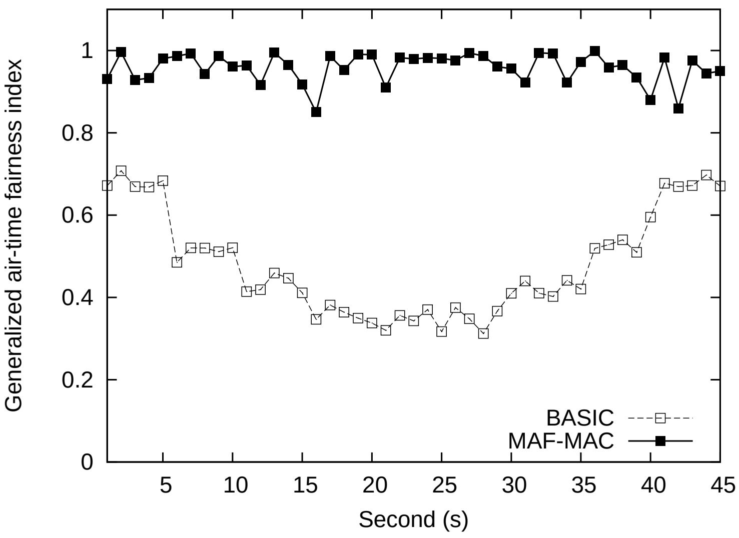
for most of the time. Figure 4.13(b) illustrates the generalized air-time fairness computed for the time interval of one second among the saturated nodes, i.e., excluding the unsaturated node in calculating the generalized air-time fairness. The saturated nodes in MAF-MAC fairly utilized the channel resource even in the presence of an unsaturated node. This simulation result shows that MAF-MAC can improve max-min air-time fairness whether there are unsaturated nodes or not in an ad-hoc network.

Short-term generalized air-time fairness

Figure 4.14 shows the short-term generalized air-time fairness for various numbers of R2 and R3 nodes, whereas the number of R1 node is zero. Regardless of the numbers of R2 and R3 nodes, BASIC could not achieve good air-time fairness, while MAF-MAC provided the generalized air-time fairness index better than 0.9 for the evaluation intervals longer than 1 second.



(a) Throughput performance



(b) Generalized air-time fairness

Figure 4.15 Transient time behavior

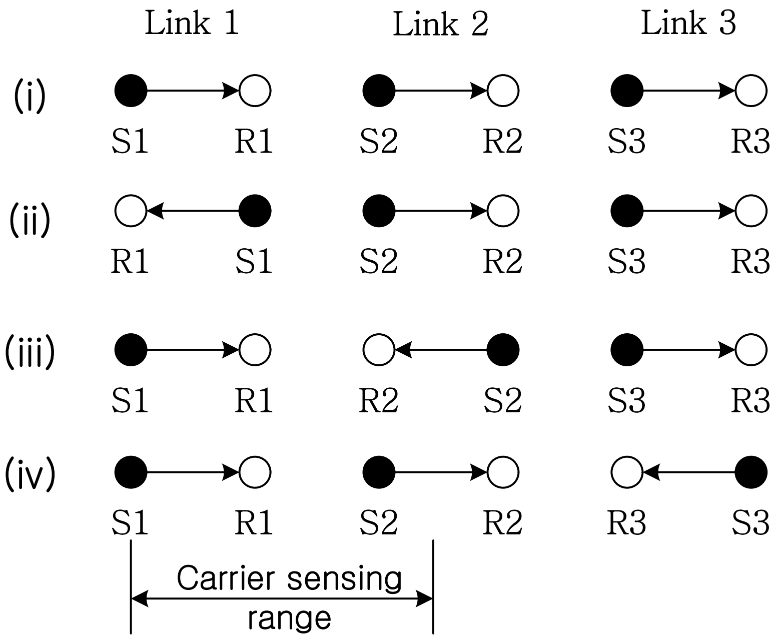


Figure 4.16 Four patterns of three flows in a line topology

Transient time behavior

Figure 4.15 shows network performance when the number of R3 nodes is fixed to two and the number of R2 nodes varies every 5 second as (1 → 2 → 3 → 4 → 5 → 4 → 3 → 2 → 1). The BASIC scheme shows poor air-time fairness despite of its higher throughput, whereas MAF-MAC provides good air-time fairness despite the frequent changes in the number of nodes in an ad-hoc network.

4.4.4 Three flows in a line topology

To show the effectiveness of MAF-MAC in improving max-min air-time fairness even when there are hidden nodes, we tested the six schemes for various flow patterns in the line topology of Fig. 4.16 , where the data rate was 26Mbps for each link. There are four different flow patterns depending on the direction of data flow. It is noted that there is no hidden node in the flow patterns (i) and (ii) of Fig. 4.16, whereas the hidden node problem occurs between

Table 4.1 Link throughput, $\overline{J}$, C_{oc}, C_{ut} and T_{del} of the six schemes in the line topologies of three flows.

Topology	Scheme	Throughput (Mb/s)				$\overline{J}$	C_{oc}	C_{ut}	T_{del} (ms)
		Link 1	Link 2	Link3	Aggregate				
Line (i)	BASIC	19.843	1.505	19.932	41.281	0.626	0.939	0.923	4.056
	BASIC(HD/E)	19.825	1.509	19.926	41.260	0.625	0.938	0.925	4.145
	BASIC(ROC/HD/E)	19.815	1.532	19.905	41.252	0.623	0.939	0.925	4.060
	MAF-MAC	13.292	7.626	13.322	34.239	0.939	0.912	0.904	1.533
	MAF(HD/E)	13.291	7.622	13.339	34.253	0.938	0.911	0.903	1.536
	MAF(ROC/HD/E)	13.189	7.693	13.225	34.108	0.942	0.912	0.905	1.534
Line (ii)	BASIC	19.815	1.485	19.864	41.194	0.627	0.939	0.925	4.402
	BASIC(HD/E)	19.846	1.445	19.929	41.221	0.621	0.940	0.924	4.332
	BASIC(ROC/HD/E)	19.875	1.417	19.957	41.251	0.623	0.938	0.923	4.453
	MAF-MAC	13.327	7.594	13.342	34.263	0.938	0.911	0.905	1.539
	MAF(HD/E)	13.347	7.572	13.341	34.259	0.937	0.912	0.905	1.533
	MAF(ROC/HD/E)	13.188	7.748	13.199	34.135	0.943	0.910	0.902	1.538
Line (iii)	BASIC	19.7294	0.006	20.528	40.263	0.695	0.958	0.615	910.746
	BASIC(HD/E)	20.194	0.051	21.018	41.262	0.673	0.908	0.658	103.828
	BASIC(ROC/HD/E)	19.825	1.105	20.057	40.989	0.710	0.938	0.901	5.617
	MAF-MAC	7.938	0.005	10.738	18.675	0.997	0.603	0.312	1092.537
	MAF(HD/E)	7.889	6.302	7.378	21.568	0.998	0.837	0.673	2.302
	MAF(ROC/HD/E)	14.458	6.232	8.871	29.561	0.891	0.851	0.823	2.178
Line (iv)	BASIC	20.498	0.007	19.686	40.193	0.697	0.957	0.613	833.876
	BASIC(HD/E)	20.679	0.051	20.208	41.235	0.673	0.907	0.659	113.168
	BASIC(ROC/HD/E)	19.485	0.125	19.538	39.147	0.676	0.911	0.616	34.672
	MAF-MAC	10.537	0.006	7.789	18.327	0.998	0.653	0.395	912.538
	MAF(HD/E)	7.421	6.199	7.906	21.527	0.997	0.837	0.632	2.315
	MAF(ROC/HD/E)	8.527	3.206	15.431	27.164	0.851	0.849	0.678	2.914

senders S1 and S2 in flow pattern (iii), and S2 and S3 in flow pattern (iv). Table 4.1 shows the link throughput, generalized air-time fairness index ($\overline{J}$), channel occupation ratio (C_{oc}), channel utilization ratio (C_{ut}) and average packet delay (T_{del}) of each scheme for the four flow patterns in Fig. 4.16, and we summarize the result as the following.

- **Flow patterns in Fig. 4.16(i) and (ii)**: There is no hidden node in flow patterns (i) and (ii), thus the network performance is similar to the previous simulation study of the starvation scenario. Although the aggregate throughputs of **BASIC**, **BASIC(HD/E)** and **BASIC(ROC/HD/E)** is higher than those of **MAF-MAC**, **MAF(HD/E)** and **MAF(ROC/HD/E)**, link2 suffered from the starvation problem. The transmissions in link1 and link3 occupied the channel most of the time, and consequently the fairness performance was severely degraded. On the other hand, **MAF-MAC**, **MAF(HD/E)** and **MAF(ROC/HD/E)** provided much better air-time fairness compared to **BASIC**, **BASIC(HD/E)** and **BASIC(ROC/HD/E)** (from approximately 0.62 to 0.93) with high C_{oc} and C_{ut} , i.e., achieved much better max-min air-time fairness in the line flow patterns (i) and (ii). Furthermore, the packet delays in **MAF-MAC**, **MAF(HD/E)** and **MAF(ROC/HD/E)** were significantly reduced compared to **BASIC**, **BASIC(HD/E)** and **BASIC(ROC/HD/E)** by resolving the starvation problem.

- **Flow pattern in Fig. 4.16(iii)**: In this flow pattern, the transmission in link1 and link2 suffered from the hidden node problem. Therefore, the air-time fairness of **BASIC**, aggregate throughput of **MAF-MAC** and packet delay performances of both schemes were severely degraded. Note that the transmission in link2 still suffers from the starvation problem in the flow pattern (iii) of Fig. 4.16, and thus link2 cannot use its fair share of the channel in the **BASIC(HD/E)** and **BASIC(ROC/HD/E)** schemes. We notice that the hidden detection mechanism with extended CTS range significantly contributes to resolve the starvation problem in **MAF(HD/E)**, i.e., link2 can successfully utilize its fair share. Moreover, **MAF(HD/E)** provides higher C_{ut} (from approximately 0.46 to 0.67) and much lower T_{del} (from approximately 1001.6ms to 2.3ms) compared to **BASIC** and **MAF-MAC**. When the ROC mechanism was applied to MAF-MAC (**MAF(ROC/HD/E)**, the air-time fairness was degraded (from approximately 0.99 to 0.89) as we expected. However, nodes R1, R2 and S3 participated in channel contention by the ROC mechanism, which can remove the hidden node problem by placing all the contending nodes in the carrier sensing range of each other. This results in the improvement of throughput and C_{ut} but degrades the air-time fairness compared to **MAF(HD/E)**.

In the flow pattern (iii), **MAF(HD/E)** and **MAF(ROC/HD/E)** show much higher the max-min air-time fairness compared to **BASIC(HD/E)** and **BASIC(ROC/HD/E)**.

- **Flow pattern in Fig. 4.16(iv)**: In this flow pattern, `link2` and `link3` suffered from the hidden node problem, while `link2` suffered from the starvation problem as the flow pattern (iii) in Fig. 4.16. When the ROC mechanism was not applied, there were both starvation and hidden node problems in the flow pattern (iv) as in the flow pattern (iii). Thus, **Basic**, **BASIC(HD/E)**, **MAF-MAC** and **MAF(HD/E)** shows nearly the same performance as their counterparts in the flow pattern (iii). When the ROC mechanism is used, `S1`, `R2` and `R3` participate in channel contention, and a hidden node problem occurs between `link1` and `link2`. Therefore, the air-time fairness of **MAF(ROC/HD/E)** is degraded compared to that of **MAF(HD/E)** (from 0.99 to 0.85). It is noted that **BASIC(ROC/HD/E)** could not properly resolve hidden node and starvation problems, and thus it provided poor air-time fairness and delay performance. Because of the hidden node introduced by the ROC mechanism, **MAF(HD/E)** showed better network performance compared to **MAF(ROC/HD/E)**. Still **MAF(ROC/HD/E)** provides better air-time fairness and delay performance with high C_{oc} and C_{ut} compared to **BASIC**, **BASIC(HD/E)** and **BASIC(ROC/HD/E)**.

For each case in Fig. 4.16, **MAF(HD/E)** and **MAF(ROC/HD/E)** successfully resolve both starvation and hidden node problems, and improve max-min air-time fairness with good delay performance compared to **BASIC**, **BASIC(HD/E)**, **BASIC(ROC/HD/E)** and **MAF-MAC**.

4.4.5 Double ring topologies

We tested the six schemes in the double ring topologies as shown in Fig. 4.17. This topologies were introduced to demonstrate the usefulness of the ROC mechanism [9]. There are two kinds of flow patterns depending on the direction of data flow, and there are two topologies, a

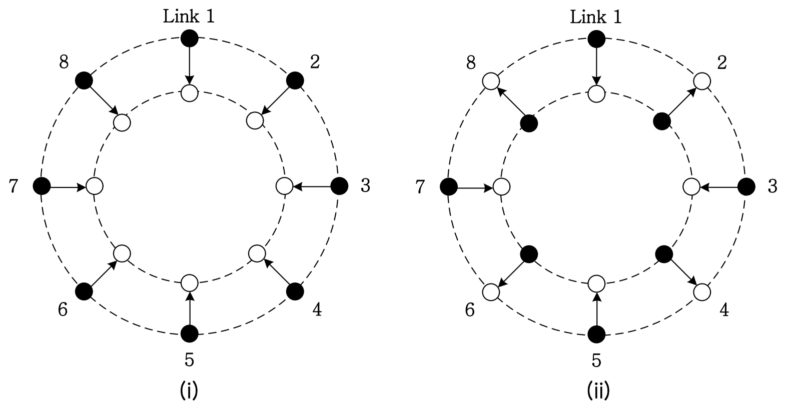


Figure 4.17 Two kinds of double ring topologies.

smaller one and a larger one for each flow pattern. The radius of inner circle is always fixed to 150 meters, and radius of outer circles for small and large topologies were 250 and 300 meters, respectively. The data rate of small and large topologies were set to 26Mbps and 19.5Mbps, respectively. It is noted that more nodes suffered from the hidden node problem in the larger topology. The hidden node problem cannot be resolved by only using the hidden detection mechanism with the extend CTS range for larger topologies. Table 4.2 shows the link throughput, $\overline{J}$, C_{oc}, C_{ut} and T_{del} of each scheme for the four topologies, and we summarize the result as the following.

- **Small double ring topology in Fig. 4.17(i)** : The senders and receivers were uniformly located in the outer and inner circles, respectively. Transmission of each link was interfered by hidden nodes, and thus, **BASIC** and **MAF-MAC** showed poor throughput performance. Note that the each link fairly utilized the channel, but the probability of successful transmission was very low, because the ACK transmission of each link was interfered by the transmissions of the other links. Thus, although $\overline{J}$ was close to 1, C_{ut} was very low compared to C_{oc} and the packet delay performance was severely degraded. The hidden detection mechanism with extended CTS range alleviated the effect of hidden nodes, and thus **BASIC(HD/E)** and **MAF(HD/E)** provided good throughput performance as shown in Table 4.2. When the ROC mechanism was applied, receivers participated in channel contention instead of senders. This effectively resolved the hidden node problem, because all the receivers could carrier sense each other. Therefore, **BASIC(ROC/HD/E)** and **MAF(ROC/HD/E)** show the best throughput and packet delay performance with good air-time fairness.

- **Large double ring topology in Fig. 4.17(i)**: In this case, there are more hidden nodes for each link. **BASIC** and **MAF-MAC** schemes showed poor network performance as in the small topology of Fig. 4.17 (i). Moreover, the hidden node problem could not be resolved by only using the the hidden detection mechanism with the extended CTS range, and thus **BASIC(HD/E)** and **MAF(HD/E)** could not provide good throughput and delay performance as shown in Table 4.2. The hidden node problem was effectively resolve only when the ROC mechanism was applied, and **BASIC(ROC/HD/E)** and **MAF(ROC/HD/E)** showed good throughput and delay performance and air-time fairness with high C_{oc} and C_{ut}.

- **Small double ring topology in Fig. 4.17(ii)**: In this case, half of the senders transmitted data frames inwards and the other half transmitted data frames outwards. The outward links (`Links` 1, 3, 5 and 7) suffered from the hidden node problem whereas the others did not. Most of the transmissions from inner senders failed because of the transmissions from outer senders, and thus the throughput and packet delay of outward links were very poor in **BASIC** and **MAF-MAC**. **BASIC(HD/E)** and **BASIC(ROC/HD/E)** somewhat alleviated the hidden node problem, however, the inward links still showed low throughput performance. Only in **MAF(HD/E)** and **MAF(ROC/HD/E)**, all the links fairly and successfully utilized the channel, and at the same time transmited data frames with short packet delay as shown in Table 4.2.

- **Large double ring topology in Fig. 4.17(ii)**: When the outer ring is larger, the hidden node problem is no longer resolved by only using the the hidden detection mechanism with the extended CTS range as in the large ring topology Fig. 4.17 (i). Therefore, the outward links suffered from the hidden node problem, and their throughput and delay performances were severely degraded in **BASIC**, **BASIC(HD/E)**, **MAF-MAC**, and **MAF(HD/E)**. When the ROC mechanism was applied, only the receivers of outward links participated in channel contention. Therefore, **BASIC(ROC/HD/E)** and **MAF(ROC/HD/E)** could effectively resolve the hidden node problem and all the links could fairly share the channel, because the receivers of outward links and the senders of inward links could carrier sense each other. Moreover, they provided the best delay performance in this topology.

In all the cases of double ring topologies, **MAF(ROC/HD/E)** makes each link effectively utilized and fairly shared the channel by resolving both the hidden node and starvation problems, resulting in good throughput, packet delay and air-time fairness performance.

4.4.6 Random topologies

Finally, we investigated the network performance of the six schemes for a duration of 20 seconds in various environments when all the nodes were randomly located in a square area of 500m×500m. We simulated networks of single data rate in noise-free channel for various link topologies, i.e., first for twenty different 20-link topologies, and then varying the number of links from 20 to 100. Then we simulated the networks again as in the above, except that data were transmitted in four different rates depending on the link. We performed the simulations once more, this time in noisy channel. For each case, the network performance was evaluated in terms of throughput, the generalized air-time fairness index, channel occupation ratio, channel

Table 4.2 Link throughput, $\overline{J}$, C_{oc}, C_{ut} and T_{del} of the six schemes in the double ring topologies.

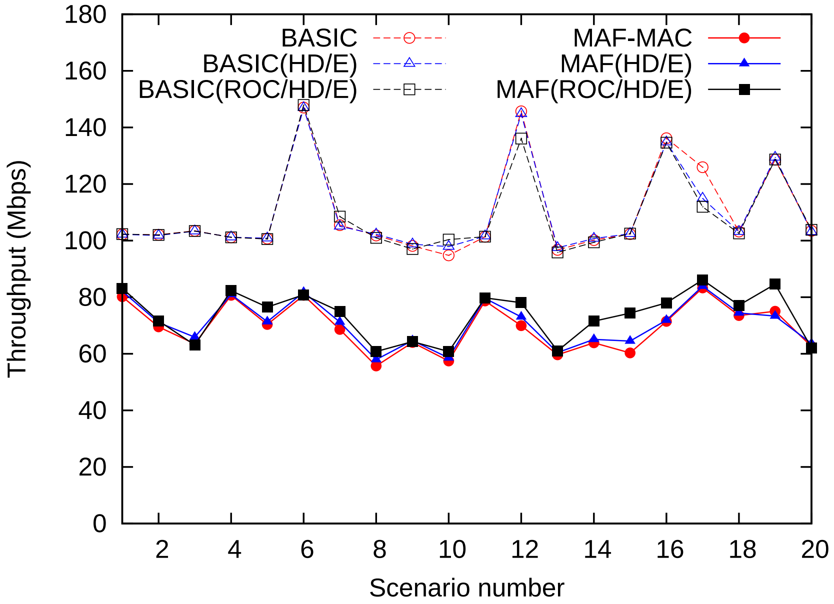
Topology	Scheme	Throughput (Mb/s)									$\overline{J}$	C_{oc}	C_{ut}	T_{del} (ms)
		Link 1	Link 2	Link3	Link 4	Link 5	Link6	Link 7	Link 8	Aggregate				
Double ring (S) (i)	BASIC	0.447	0.499	0.496	0.471	0.481	0.404	0.501	0.460	3.763	0.993	0.823	0.012	35.357
	BASIC(HD/E)	2.485	2.283	1.963	2.296	2.259	2.689	2.613	2.322	18.908	0.994	0.927	0.743	6.961
	BASIC(ROC/HD/E)	2.264	2.636	2.778	2.528	2.421	2.732	2.682	2.809	20.852	0.995	0.979	0.758	6.328
	MAF-MAC	0.068	0.057	0.032	0.058	0.024	0.013	0.052	0.085	0.604	0.389	0.951	0.008	217.014
	MAF(HD/E)	2.612	2.302	2.318	2.475	2.298	2.308	2.322	2.166	18.802	0.997	0.932	0.728	6.963
	MAF(ROC/HD/E)	3.083	2.805	2.363	2.495	2.506	2.811	2.647	2.765	21.478	0.995	0.985	0.769	6.143
Double ring (L) (i)	BASIC	0.963	0.945	0.892	0.995	0.928	0.909	1.023	0.993	7.612	0.999	0.824	0.096	17.249
	BASIC(HD/E)	1.047	0.831	0.859	0.898	1.013	0.946	0.911	0.802	7.235	0.998	0.772	0.153	18.196
	BASIC(ROC/HD/E)	1.925	1.894	1.681	1.884	2.162	1.986	1.745	1.961	15.238	0.996	0.979	0.728	8.507
	MAF-MAC	0.159	0.172	0.189	0.156	0.129	0.188	0.194	0.159	1.347	0.999	0.928	0.027	100.932
	MAF(HD/E)	0.216	0.219	0.231	0.250	0.216	0.248	0.243	0.242	1.861	0.999	0.889	0.117	71.336
	MAF(ROC/HD/E)	1.812	1.832	1.875	1.834	1.910	1.892	1.796	1.898	14.841	0.999	0.982	0.727	8.832
Double ring (S) (ii)	BASIC	0.164	5.063	0.147	4.977	0.091	4.936	0.139	4.993	20.508	0.699	0.978	0.484	16.641
	BASIC(HD/E)	1.187	3.808	1.012	3.547	1.661	3.359	1.708	3.763	20.044	0.828	0.967	0.743	10.096
	BASIC(ROC/HD/E)	1.787	3.354	1.789	2.986	2.109	2.836	1.996	3.072	19.929	0.949	0.978	0.757	6.864
	MAF-MAC	0.178	2.928	0.185	2.903	0.144	2.678	0.194	2.966	12.176	0.968	0.942	0.469	47.343
	MAF(HD/E)	2.453	2.853	2.495	2.627	2.253	2.707	2.391	2.567	20.627	0.997	0.948	0.881	6.391
	MAF(ROC/HD/E)	2.797	2.123	3.239	2.087	2.851	2.027	3.201	2.121	20.547	0.989	0.982	0.785	6.642
Double ring (L) (ii)	BASIC	0.002	4.181	0.005	4.065	0.002	4.206	0.001	4.099	16.561	0.987	0.889	0.471	4094.242
	BASIC(HD/E)	0.129	4.109	0.167	3.987	0.145	3.889	0.161	3.971	16.561	0.736	0.866	0.512	58.304
	BASIC(ROC/HD/E)	1.858	2.384	1.553	2.507	1.517	2.510	1.814	2.394	16.534	0.974	0.986	0.803	8.231
	MAF-MAC	0.004	3.748	0.003	3.778	0.003	3.817	0.004	3.685	15.039	0.991	0.832	0.464	2653.696
	MAF(HD/E)	0.434	2.821	0.393	2.829	0.348	2.872	0.421	2.791	12.912	0.986	0.831	0.435	23.568
	MAF(ROC/HD/E)	2.211	1.847	2.225	1.728	2.186	1.788	2.257	1.824	16.068	0.983	0.983	0.836	8.252

occupation ratio and packet delay. Note that the hidden node and starvation problems may coexist and theses can result in significant degradation of network performance in such complex network topologies.

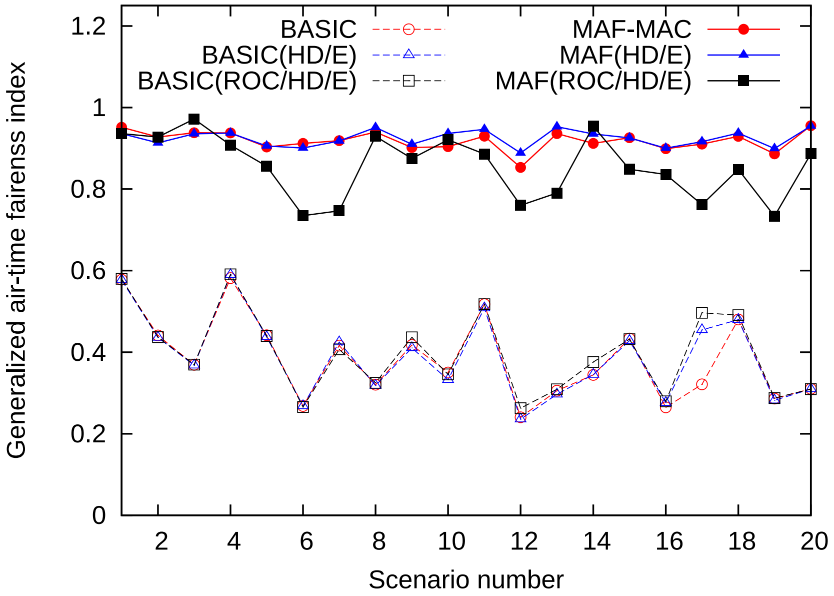
Single data rate in noise free channel

We tested the six schemes with twenty different random topologies of 20 links, and their performances are shown in Fig. 4.18. Although **BASIC**, **BASIC(HD/E)** and **BASIC(ROC/HD/E)** show higher throughput performance compared to **MAF-MAC**, **MAF(HD/E)** and **MAF(ROC/HD/E)**, their performance of air-time fairness are poor. It means that there always were starving links because of the hidden node and/or starvation problems, and some links monopolized the channel. The schemes based on MAF-MAC shows good air-time fairness in all the cases. Also, the channel was fully and effectively used in **MAF-MAC**, **MAF(HD/E)** and **MAF(ROC/HD/E)**. Figure 4.18(e) shows the average packet delay of each scheme. (The average packet delay that is longer than 500ms is not shown in Fig. 4.18(e).) **BASIC**, **BASIC(HD/E)** and **BASIC(ROC/HD/E)** provided very poor delay performance (from 50 to 5000ms), because there always were starving nodes. On the contrary, there was no starving nodes in the schemes based on MAF-MAC, which gave shorter average packet delay (mostly less than 10ms) for all the topologies. There are not much performance differences among the schemes based on MAF-MAC.

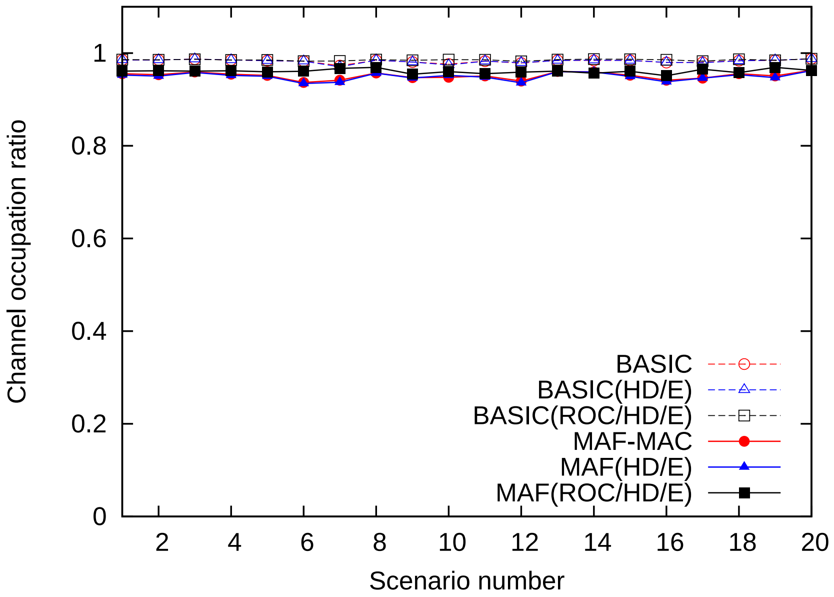
Figure 4.19 shows the average network performances for thirty random topologies as the number of links increases. The throughput performances of the schemes based on BASIC achieved higher throughput than those based on MAF-MAC regardless of the number of links. However, there always were starving nodes in the networks for the schemes based on BASIC, resulting in poor air-time fairness and delay performance. (We do not present the average packet delay that is longer than 3000ms in Fig. 4.19(e).) When the ROC mechanism was applied, the air-time fairness of **MAF(ROC/HD/E)** degraded compared to those of **MAF-MAC** and



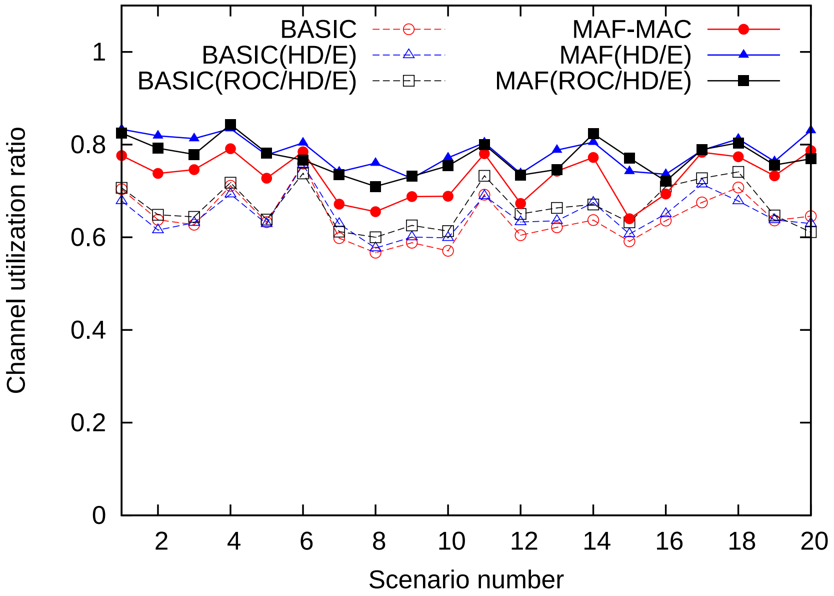
(a) Throughput performance



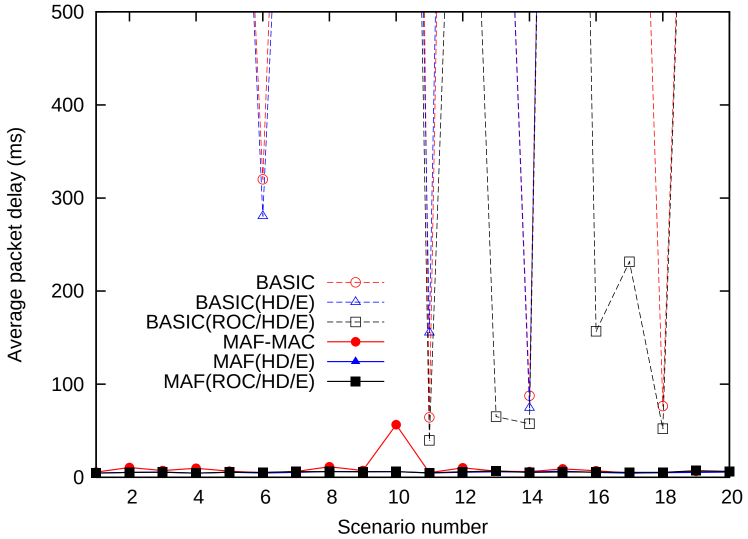
(b) Generalized air-time fairness



(c) Channel occupation ratio



(d) Channel utilization ratio



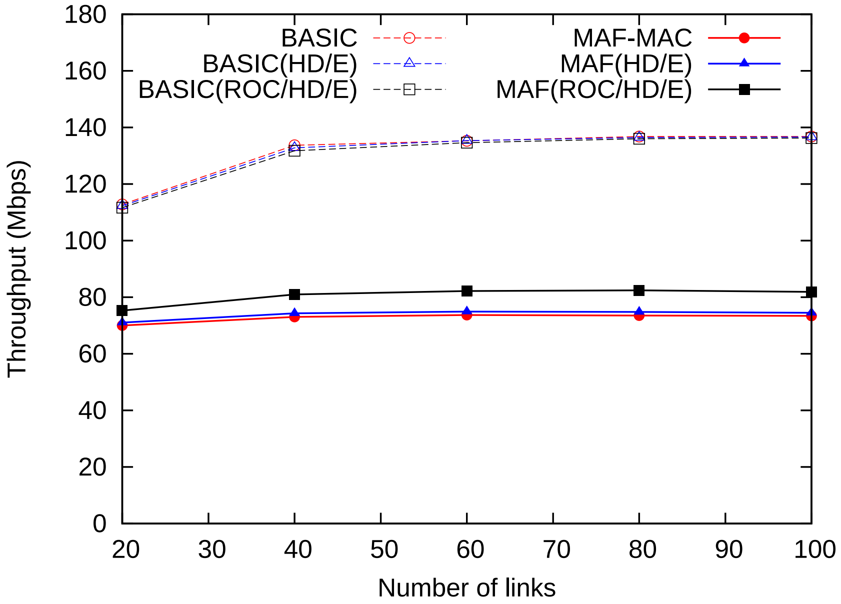
(e) Average packet delay

Figure 4.18 Network performance for single data rate in noise free channel

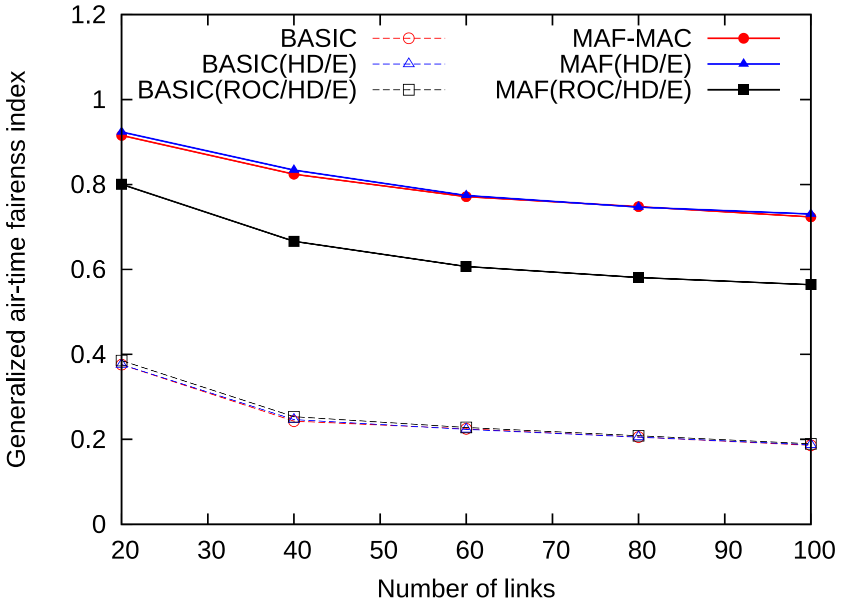
MAF(HD/E). This is because the receiver that initiated a transmission could only announce the previous value of $\widehat{\alpha}$ and had to contend based on the binary exponential backoff (BEB) mechanism. However, **MAF(ROC/HD/E)** still shows much better air-time fairness compared to the schemes based on BASIC regardless of number of links. Moreover, the schemes based on MAF-MAC more effectively utilized the channel compared to those based on BASIC as shown in Fig. 4.19(d). The average packet delay for the schemes based on MAF-MAC increased much slowly compared to those based on BASIC. As in the previous case, the schemes based on MAF-MAC effectively resolved the hidden node and starvation problems, and thus significantly improved max-min air-time fairness regardless of the number of links.

Multi data rate in noise free channel

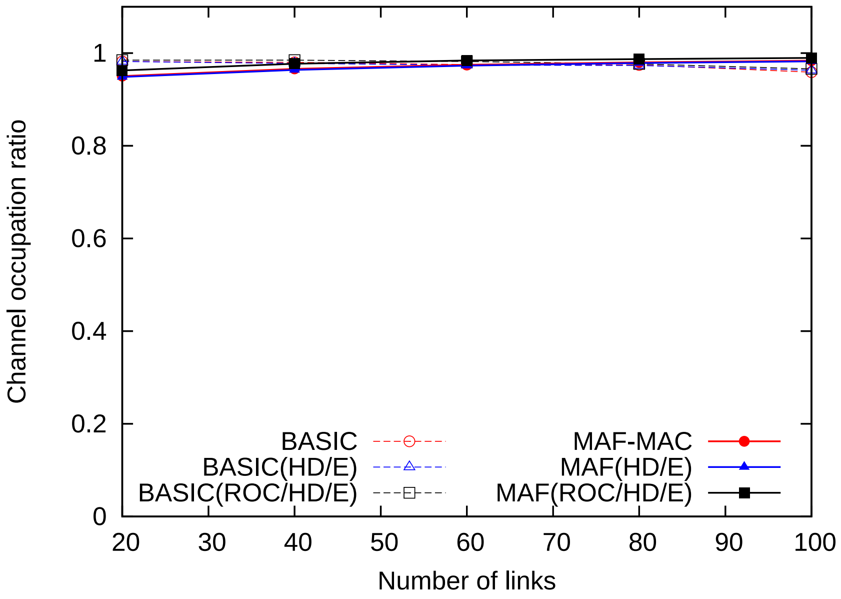
We tested the six schemes for multi data rate networks in noise free channel, and Fig. 4.20 shows various network performance for 20 networks of random topology. There were five links



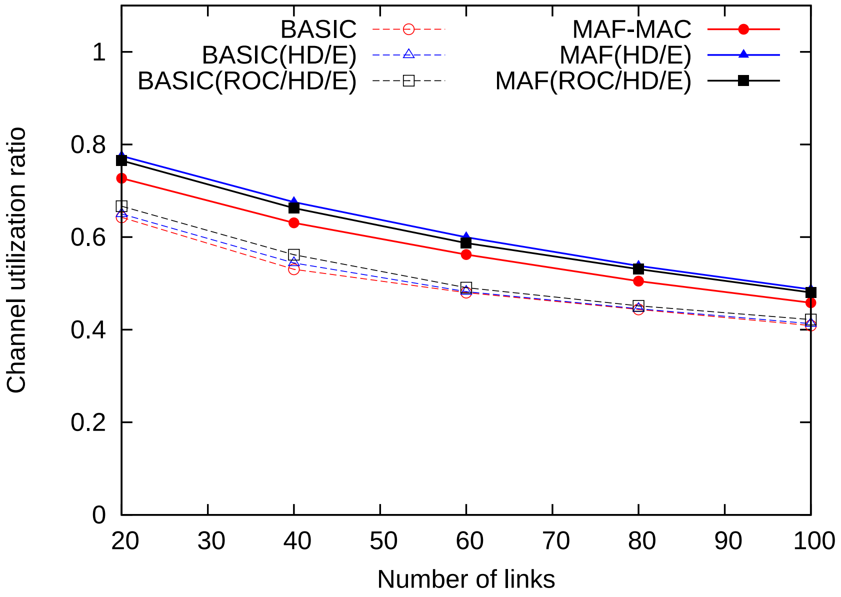
(a) Throughput performance



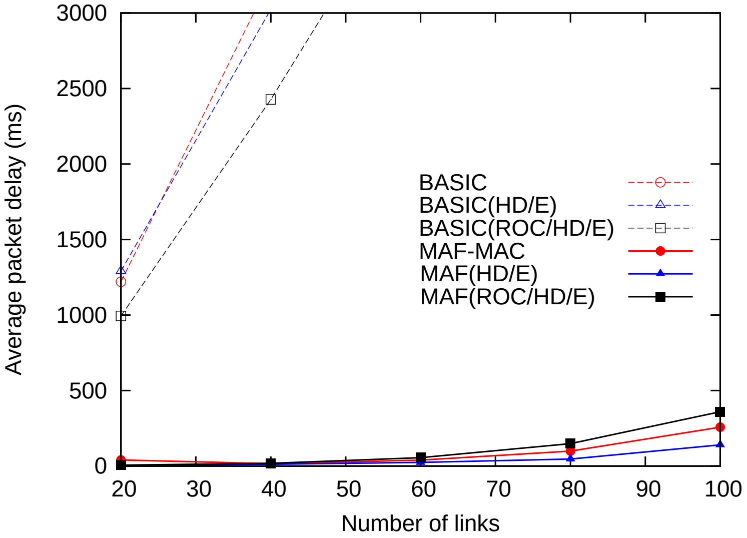
(b) Generalized air-time fairness



(c) Channel occupation ratio



(d) Channel utilization ratio

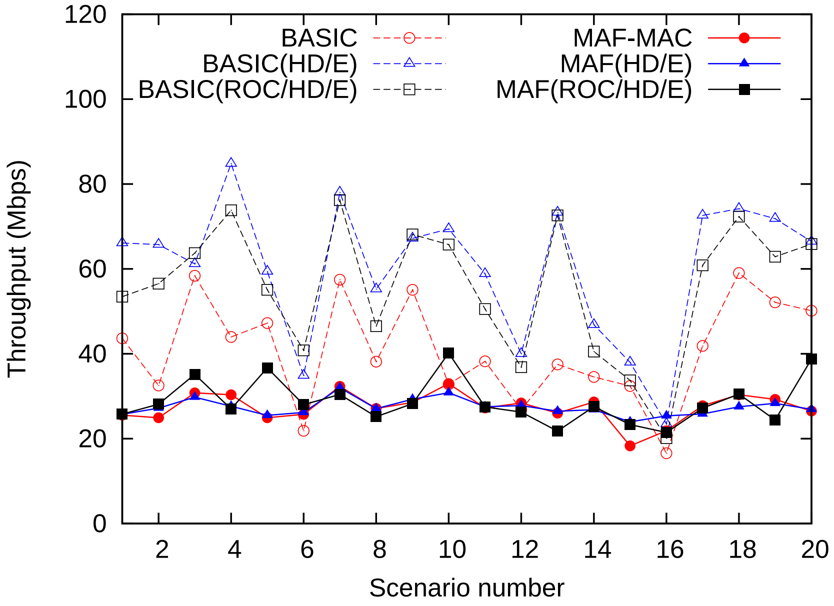


(e) Average packet delay

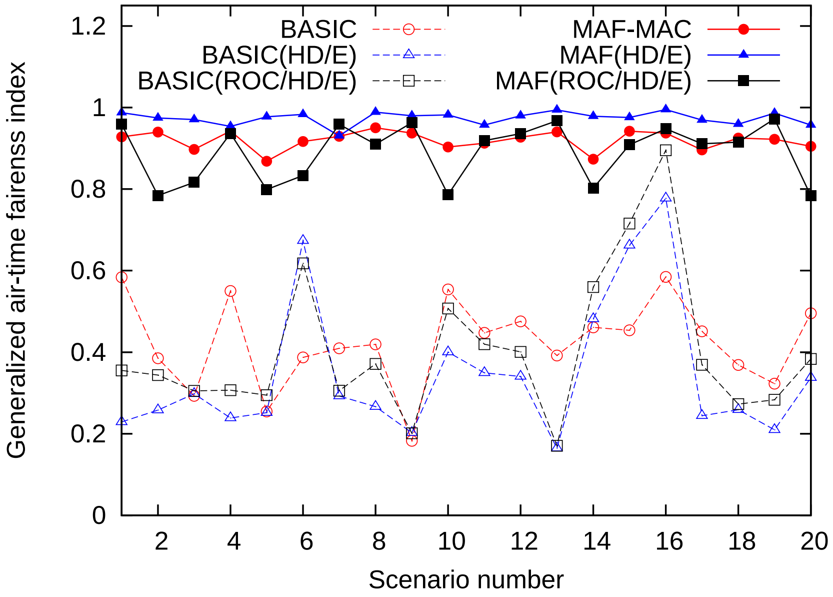
Figure 4.19 Network performance for various number of links in single data rate networks

for each of the data rates of 6.5, 13, 52 and 65Mbps. Since the link distance was longer for a lower data rate, the probability that a low rate link suffered from the hidden node problem increased. This is because there are more possibilities of encountering nodes that are out of the carrier sensing range of a sender, but can still interfere data frame reception of the receiver. For this reason, the hidden node problem occurred more frequently in multi-rate environment than a single data rate of 65Mbps network.

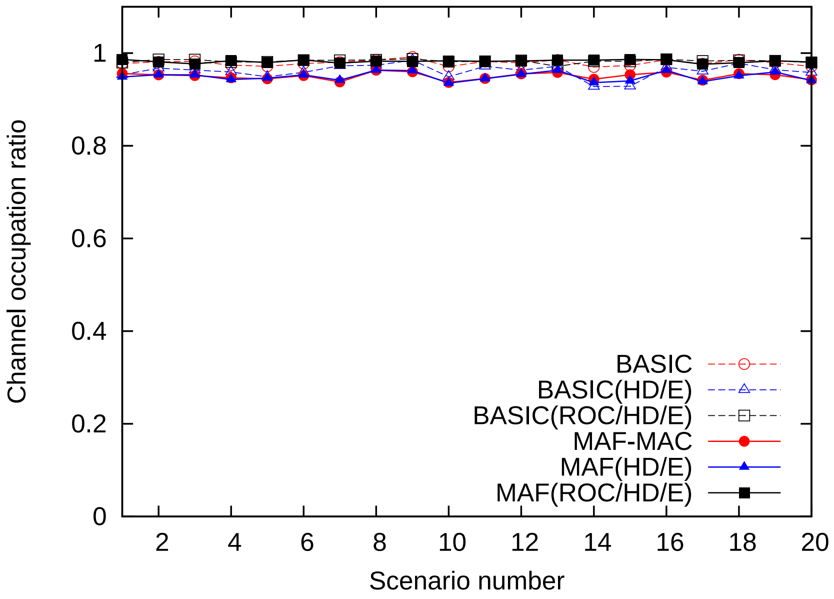
It is noted that **BASIC** has no features to improve air-time fairness in multi-rate environment, whereas **BASIC(HD/E)** and **BASIC(ROC/HD/E)** both use the frame size adaptation scheme to extend the effective CTS range that was proposed to improve air-time fairness in multi-rate environment [9]. Although **BASIC(HD/E)** and **BASIC(ROC/HD/E)** show higher throughput performance compared to **BASIC** by somewhat alleviating the performance anomaly, they still show poor air-time fairness because they could not resolve the starvation problem as we can see in 4.20(b). This implies that the air-time fairness cannot be improved by



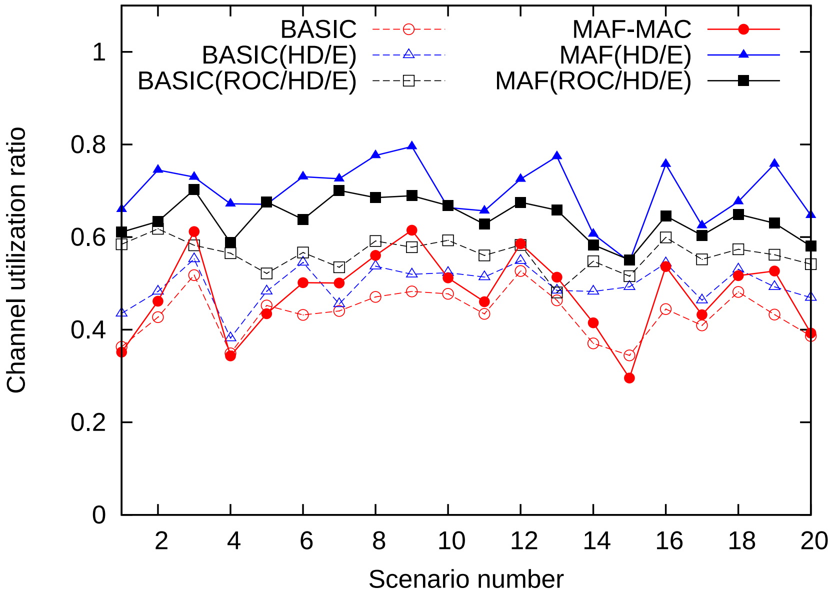
(a) Throughput performance



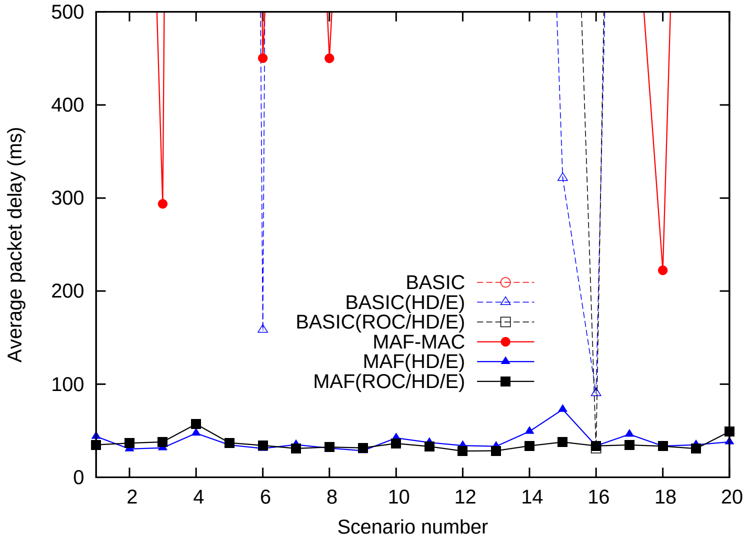
(b) Generalized air-time fairness



(c) Channel occupation ratio



(d) Channel utilization ratio

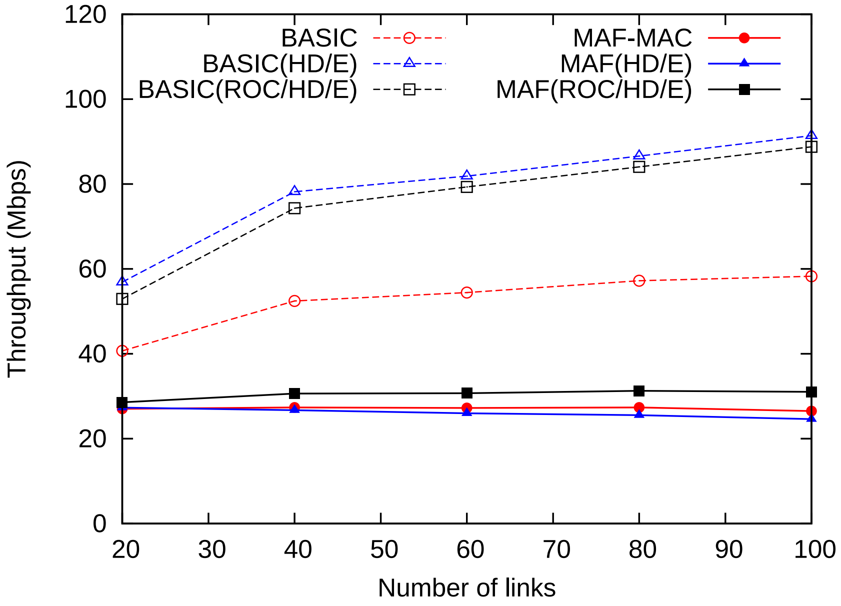


(e) Average packet delay

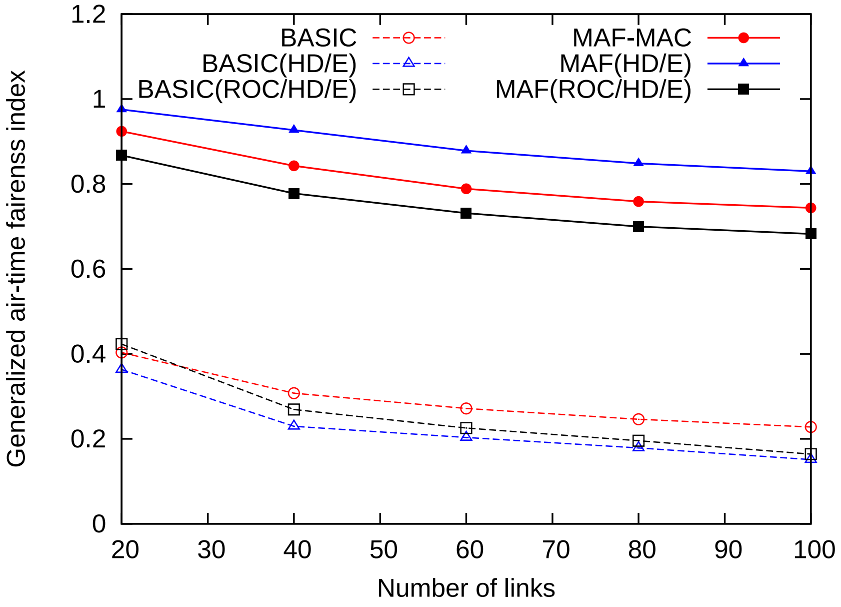
Figure 4.20 Network performance for multi data rates in noise free channel

only using the frame size adaptation in ad-hoc networks.

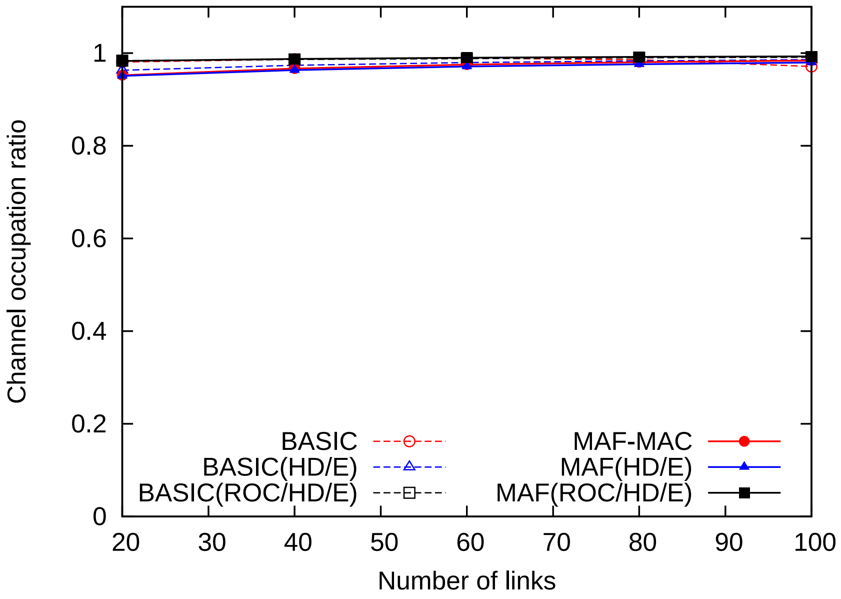
In **MAF-MAC**, the air-time fairness was significantly improved compared to the schemes based on BASIC. However, the channel utilization ratio and delay performance were severely degraded as shown in Fig. 4.20(d) and 4.20(e), because **MAF-MAC** could not resolve the hidden node problem, and thus most of transmissions interfered by hidden nodes failed. (We do not present the average packet delay that is longer than 500ms in Fig. 4.20(e).) **MAF(HD/E)** and **MAF(ROC/HD/E)** provided good air-time fairness and delay performance even though the throughput was lower than the schemes based on BASIC. This does not mean that the channels were not well utilized as can be seen by the high channel occupation ratio and channel utilization ratio, but implies that channel was used by lower rate links as well as higher rate links. There is not much difference in the network performance between **MAF(HD/E)** and **MAF(ROC/HD/E)** that successfully resolve both the hidden node and starvation problems in multi data rate networks.



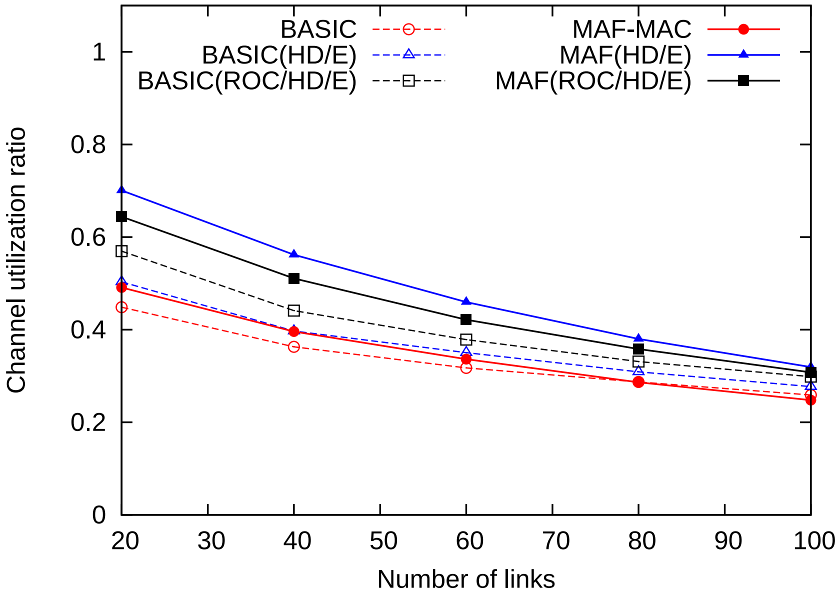
(a) Throughput performance



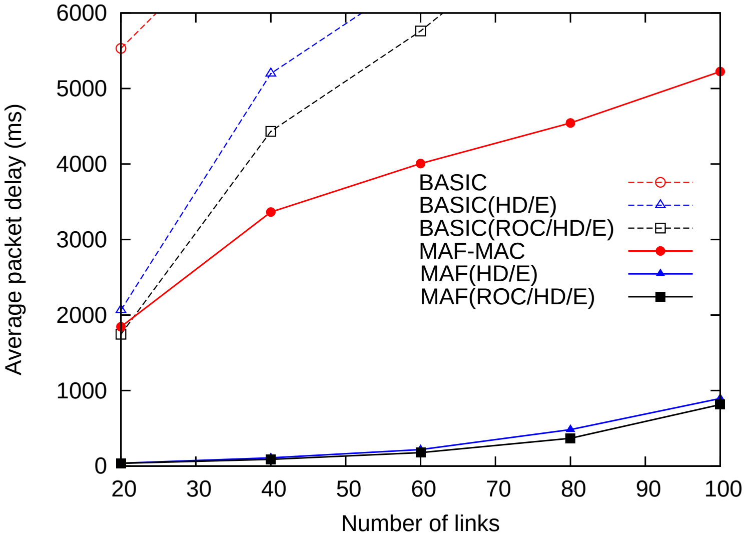
(b) Generalized air-time fairness



(c) Channel occupation ratio



(d) Channel utilization ratio



(e) Average packet delay

Figure 4.21 Network performance for various number of links in multi data rate networks

Figure 4.21 shows the average network performance of thirty multi data rate random topologies for various number of links. The number of links were the same for each of the data rates (6.5, 13, 52 and 65 Mbps). And, each network performance is the average of thirty different multi-rate random topologies corresponding to a given number of links. Although the schemes based on BASIC show higher throughput performance compared to the schemes based on MAF-MAC, they provide poor air-time fairness regardless of the number of links, as can be seen in Fig. 4.21(a) and 4.21(b). It due to the hidden node and/or starvation problems, and some links could not use the channel, which can be seen by the delay performance in Fig. 4.21(e). (We do not present the average packet delay that is longer than 6000ms in Fig. 4.21(e).)

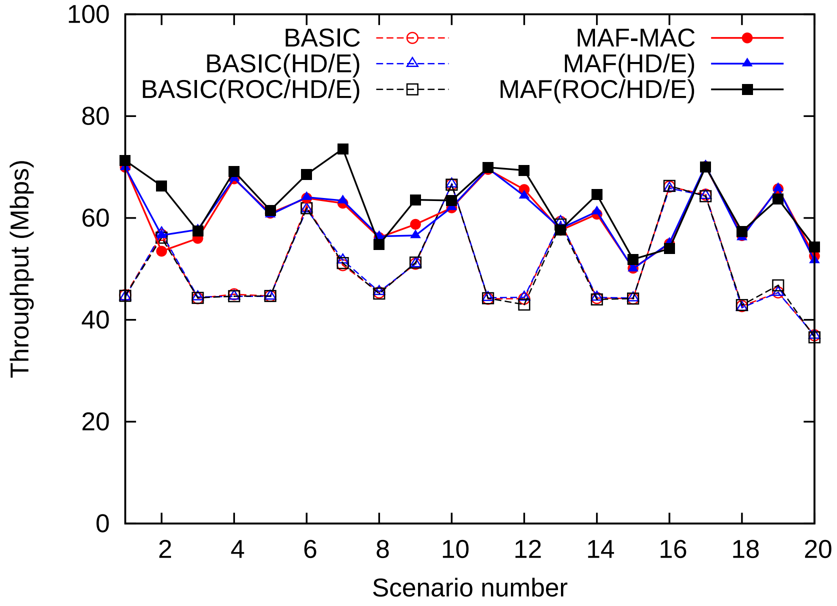
MAF-MAC provides better air-time fairness compared to the schemes based on BASIC, but it shows low channel utilization and poor delay performances compared to **MAF(HD/E)** and **MAF(ROC/HD/E)**, as can be seen in Fig. 4.21(d) and 4.21(e). This simulation results show that **MAF(HD/E)** and **MAF(ROC/HD/E)** improved max-min air-time fairness in multi data rate random topologies while achieving good throughput by resolving the hidden node and starvation problems.

Single data rate in noisy channel

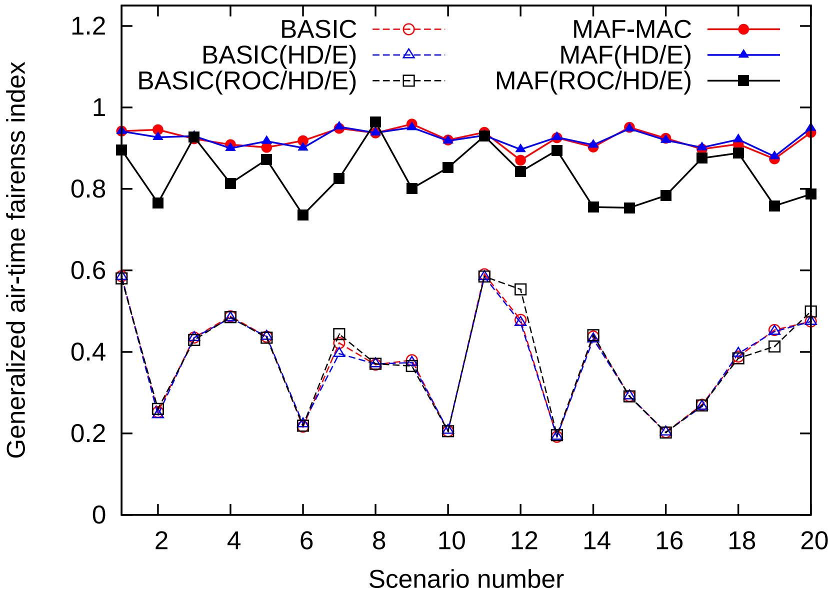
Figure 4.22 shows the network performance of single data rate networks of twenty links in noisy channel, where the bit error rate (BER) was set to $5 \cdot 10^{-5}$. Figure 4.22(d) shows that the channel utilization ratio was very low in **BASIC**, **BASIC(HD/E)** and **BASIC(ROC/HD/E)** due to the channel noise, which resulted in significant throughput degradation compared to the case of noise free channel (see Fig. 4.18(a)). Because packets were divided into fragments of 256byte, which are much shorter than the packet size of 2048byte, in the MAF-MAC based schemes, they showed higher channel utilization ratio compared to the schemes based on the BASIC as can be seen in Fig 4.22(d). In this case, the schemes based on MAF-MAC provided similar or higher throughput performance compared to the schemes based on BASIC as shown in Fig. 4.22(a). Furthermore, they also showed much better air-time fairness and delay performance compared to the schemes based on BASIC while fully utilizing the channel. (We do not present the average packet delay that is longer than 500ms in Fig. 4.22(e).) **MAF-MAC** was a little bit lower in the channel utilization ratio and **MAF(ROC/HD/E)** is a little bit lower in the generalized air-time fairness among the schemes based on MAF-MAC.

Multi data rate in noisy channel

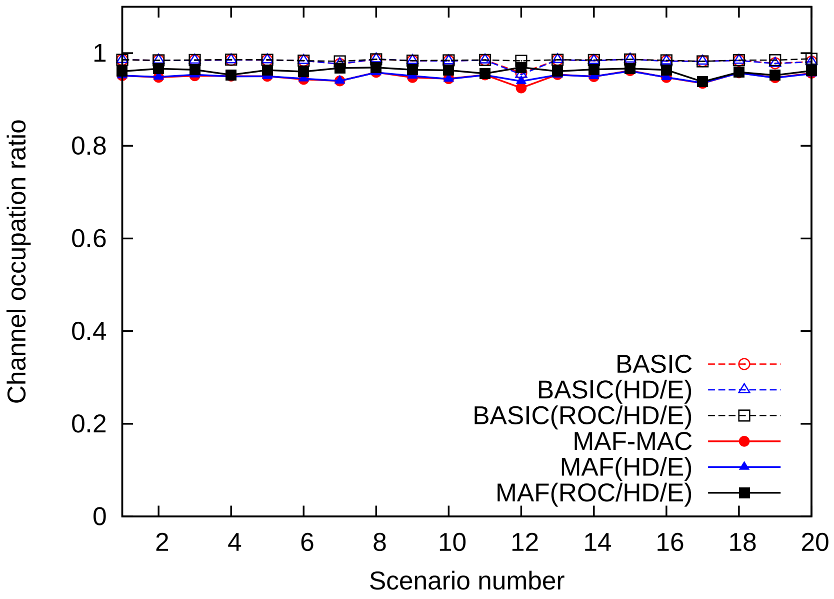
We investigate the network performance of each scheme in the same environments as in the case of multi data rate in noise free channel except that the BER of the channel was set to $5 \cdot 10^{-5}$. The throughput performance for the schemes based on BASIC in noisy channel degraded significantly compared to the case in noise free channel, whereas we can see that there are little throughput difference between the noise free channel and noisy channel in MAF-MAC based



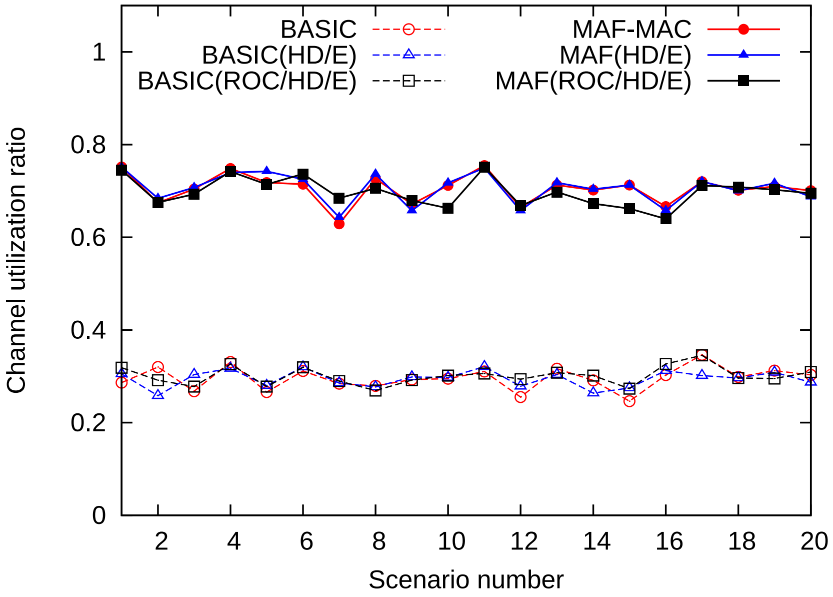
(a) Throughput performance



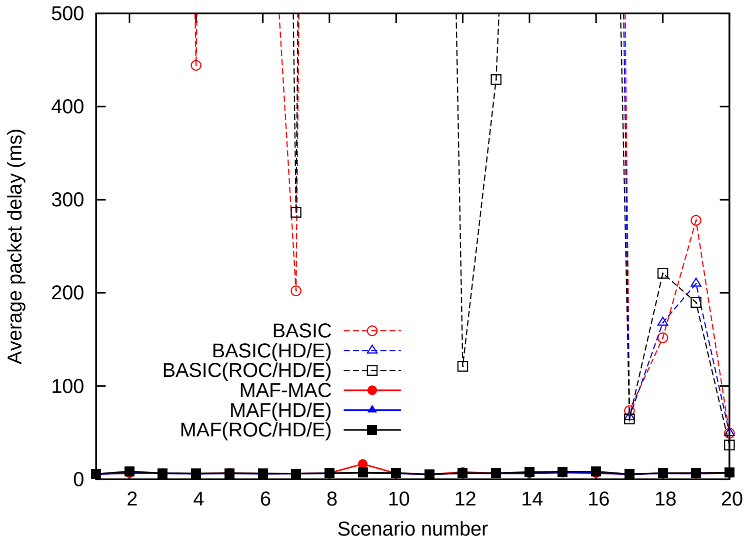
(b) Generalized air-time fairness



(c) Channel occupation ratio



(d) Channel utilization ratio

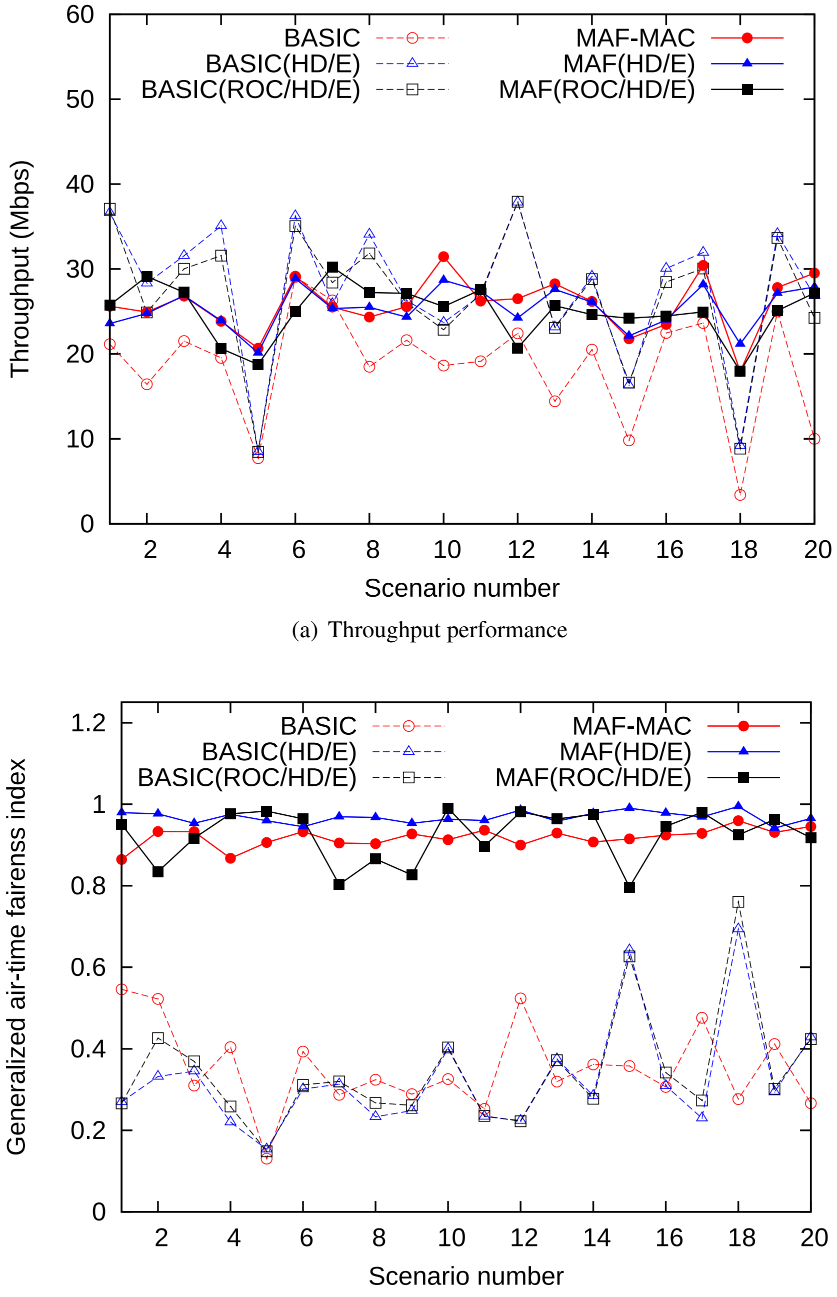


(e) Average packet delay

Figure 4.22 Network performance of single data rate networks in noisy channel

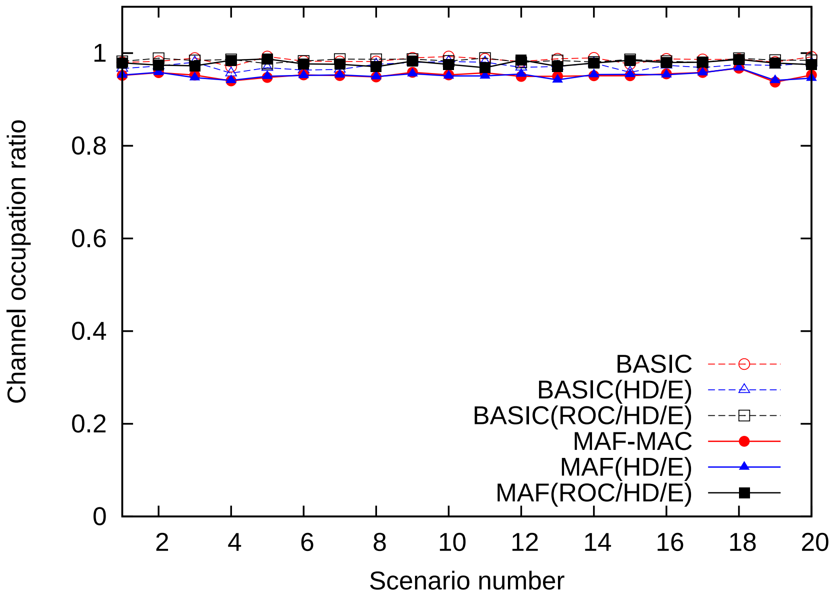
schemes by comparing Fig. 4.20(a) and Fig. 4.23(a). Note that **BASIC** provided poor throughput performance compared to **BASIC(HD/E)** and **BASIC(ROC/HD/E)** due to the performance anomaly. Moreover, Fig. 4.23(b) and 4.23(e) show that the schemes based on BASIC provided poor air-time fairness and delay performance. (We do not present the average packet delay that is longer than 500ms in Fig. 4.23(e).)

The schemes based on MAF-MAC show similar or higher throughput performance, much better air-time fairness, and higher channel utilization ratio compared to the schemes based on BASIC. As mentioned earlier, the hidden node problem occurs more frequently in this multi data rate environment than in the single data rate (65Mbps) environment. Thus, channel utilization and delay performance of **MAF-MAC** degraded because it could not effectively handle the hidden node problem. On the contrary, **MAF(HD/E)** and **MAF(ROC/HD/E)** provided much better air-time fairness and delay performance, high channel occupation ratio and channel utilization ratio for each network topology, as can be seen in Fig 4.23.

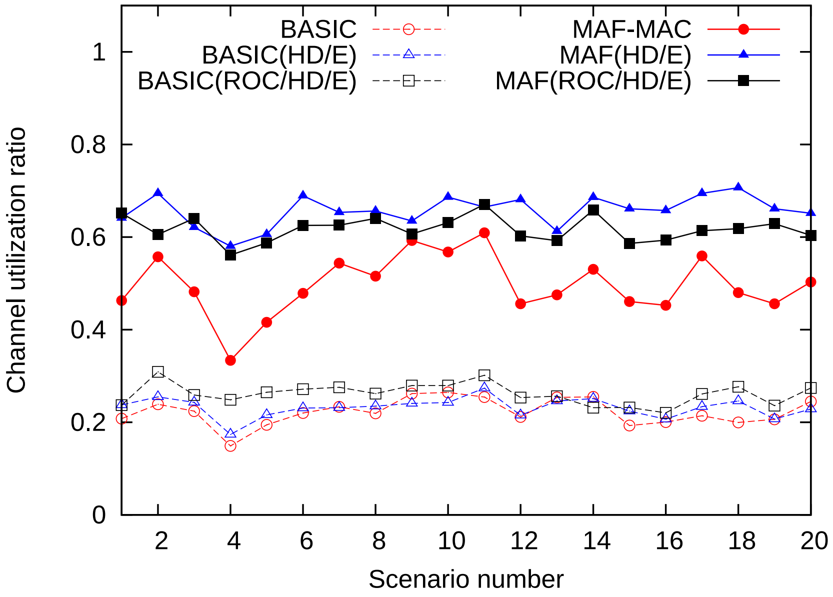


(a) Throughput performance

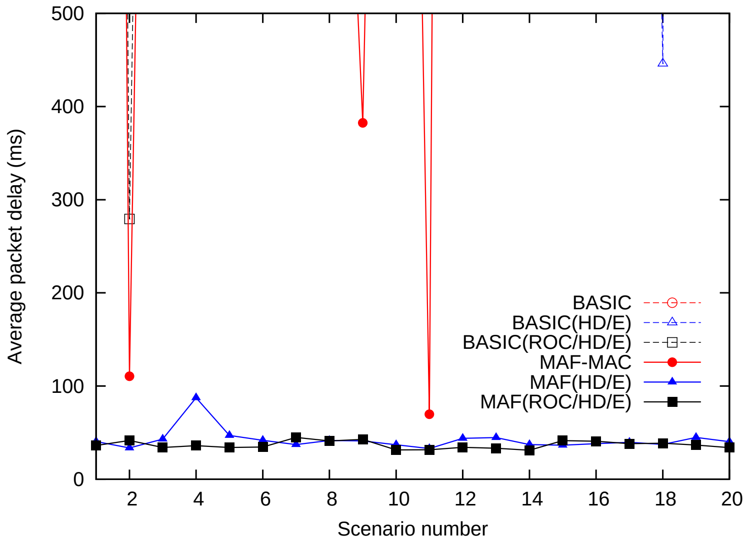
(b) Generalized air-time fairness



(c) Channel occupation ratio



(d) Channel utilization ratio



(e) Average packet delay

Figure 4.23 Network performance of single data rate networks in noisy channel

The extensive simulation results demonstrates that **MAF(HD/E)** and **MAF(ROC/HD/E)** show good network performance in various environments by effectively resolving the hidden node and starvation problems. Since **MAF(HD/E)** shows a little bit more consistent performance than **MAF(ROC/HD/E)** except for the cases of double ring topologies, it is a good idea to use **MAF(HD/E)** as a MAC protocol in ad-hoc networks.

4.5 Chapter summary

In this chapter, we proposed MAF-MAC to improve max-min air-time fairness in IEEE 802.11 ad-hoc networks while providing good throughput performance. In MAF-MAC, the transmission duration is adjusted to announce the busy time ratio $\widehat{\alpha}$ without using control messages. On the basis of the information of $\widehat{\alpha}$, each node can adjust its CW value to improve max-min air-time fairness. Moreover, by adopting the hidden node detection and resolving mechanism, MAF-MAC provides good air-time fairness and small packet delay with high channel utilization ratio even when there are hidden nodes in the network. The basic idea of proposed MAC protocol can be applied not only to WLANs, but also CSMA/CA networks.

Chapter 5

Conclusion

The MAC efficiency of IEEE 802.11 rapidly decreases as the transmission rate becomes higher because of the MAC-layer overheads such as the MAC header, contention time, and ACK transmission time. To improve the MAC efficiency, IEEE 802.11n [1] introduces several mechanisms including frame aggregation and Block ACK. However, the MAC efficiency does not improve as much as expected because of transmission collisions and channel impairment. Moreover, the IEEE 802.11 DCF is based on the CSMA/CA and BEB mechanism, and thus it cannot provide the nodes in an ad-hoc network a fair opportunity to access the channel. In order to overcome this shortcoming, we first show how each node can adjust its the transmission duration based on the frame aggregation and block ACK features of IEEE 802.11n. The transmission duration is adjusted to play the role usually carried out by control messages, and a node can indirectly announce its present state to the other nodes without incurring any overhead. Furthermore, the nodes that are in the carrier sensing range of each other can exchange information of their present state by using transmission duration. This is impossible for the schemes using a control message or an optional field in the PHY/MAC headers. This idea is simple, but very effective to enhance the network performance by exchanging necessary information without overheads. Based on this idea, we propose two MAC protocols (TOD-MAC and MAF-MAC) for enhancing network performance.

We firstly introduced the Transmission Order Deducing MAC (TOD-MAC) protocol to improve the MAC layer efficiency in a IEEE 802.11n single-cell network. As the PHY rate increases, the time to transmit a frame is quickly dominated by a fixed overhead associated with the PHY header, contention time, etc. The wasted time caused by collisions or channel errors is crucial in improving the MAC efficiency. Thus, data transmission in a round robin manner, instead of contention for an opportunity for data transmission, is an attractive alternative. If each node transmits in a round robin manner, the contention time and collision rate can both be minimized at the same time, and consequently the MAC efficiency can be significantly improved. In TOD-MAC, the transmission duration is adjusted and it performs the function of a control message to determine the transmission order of nodes. Based on the information of transmission order, each node transmits in a round robin manner, which minimizes the idle time between two consecutive transmissions and also prevents transmission collisions. The simulation results indicate that TOD-MAC not only achieves high throughput performance, but it also provides good short/long term air-time fairness and fast transient response in various dynamic environments.

We also proposed another MAC (MAF-MAC) to improve max-min air-time fairness in IEEE 802.11 ad-hoc networks. Because each node operates based on CSMA/CA in IEEE 802.11 WLANs, a node that senses channel busy for all the time never has an opportunity to transmit a data frame. Therefore, some nodes may starve and other nodes may monopolize the channel depending on their relative position in an ad-hoc network. In MAF-MAC, the transmission duration is adjusted to announce the busy time ratio $\widehat{\alpha}$ to improve max-min air-time fairness. Based on $\widehat{\alpha}$, each node can properly adjust its CW value. We also consider the hidden node problem in ad-hoc networks, because the probability that there are hidden nodes in an ad-hoc network, which can significantly degrade throughput as well as fairness performance. To alleviate the hidden node problem, we adopt the hidden node detection [8] and resolving mechanism [9] to MAF-MAC (MAF(HD/E) and MAF(HD/E/ROC)) to enhance the air-time fairness even when there are hidden nodes in ad-hoc networks. Extensive simulation results show that MAF(HD/E) and MAF(HD/E/ROC) provide good air-time fairness while fully and effectively using the channel in various environments, regardless of the number of links, network topologies, multi-data rates, channel noise. These schemes can solve the anomaly problem naturally. Moreover, the basic principle underlying the proposed MAC protocols is simple, which makes it easy to apply not only to WLANs but also to any CSMA/CA networks.

Bibliography

[1] IEEE 802.11 WG, "IEEE 802.11n: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput," *IEEE Std.*, Oct. 2009.

[2] E. Ong, J. Kneckt, O. Alanen, Z. Chang, T. Huovinen, and T. Nihtila, "IEEE 802.11 ac: Enhancements for very high throughput WLANs," in *IEEE 22nd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2011, pp. 849–853.

[3] L. Zhendong, "802.11 ac/ad: the Next Generation WLAN Technology Standard," *Modern Science & Technology of Telecommunications*, vol. 12, p. 004, 2010.

[4] T. Li, Q. Ni, D. Malone, D. Leith, Y. Xiao, and T. Turletti, "Aggregation with Fragment Retransmission for Very High-speed WLANs," *IEEE/ACM Transactions on Networking*, vol. 17, no. 2, pp. 591–604, 2009.

[5] Y. Xiao and J. Rosdahl, "Performance Analysis and Enhancement for the Current and Future IEEE 802.11 MAC Protocols," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 7, no. 2, pp. 6–19, 2003.

[6] P. Ng, S. Liew, K. Sha, and W. To, "Experimental Study of Hidden Node Problem in IEEE 802.11 Wireless Networks," *ACM SIGCOMM 2005*, Philadelphia, Pennsylvania, USA, Aug. 22 – 26, 2005.

[7] K. Hung and B. Bensaou, “Distributed Rate Control and Contention Resolution in Multicell IEEE 802.11 WLANs with Hidden Terminals,” *ACM MobiHoc 2010*, Chicago, Illinois, USA, Sep. 20 – 24, 2010.

[8] M. Kim and C.-H. Choi, “Hidden Node Detection in IEEE 802.11 n Wireless LANs,” *IEEE Transactions on Vehicular Technology*, Accepted in Feb. 2013.

[9] M. Kim, “Resolving Performance Anomaly and Hidden Node Problem in IEEE 802.11n Wirless LANs,” *Ph.D dissertation, Seoul National University*, 2012.

[10] V. Erceg *et al.*, “TGn channel models,” *IEEE 802.11 document 03/940r4*, May 2004.

[11] S. Sen, R. Roy Choudhury, and S. Nelakuditi, “CSMA/CN: Carrier Sense Multiple Access with Collision Notification,” *IEEE/ACM Transactions on Networking*, vol. 20, no. 2, pp. 544–556, 2012.

[12] E. Magistretti, K. Chintalapudi, B. Radunovic, and R. Ramjee, “Wifi-nano: Reclaiming Wifi Efficiency Through 800ns Slots,” *ACM MOBICOM 2011*, Las Vegas, Nevada, USA, Sep. 19 – 23, 2011.

[13] J. Choi, J. Yoo, S. Choi, and C. Kim, “EBA: An Enhancement of the IEEE 802.11 DCF via Distributed Reservation,” *IEEE Transactions on Mobile Computing*, vol. 4, no. 4, pp. 378–390, 2005.

[14] X. Yang and N. Vaidya, “A Wireless MAC Protocol Using Implicit Pipelining,” *IEEE Transactions on Mobile Computing*, vol. 5, no. 3, pp. 258–273, 2006.

[15] IEEE 802.11 WG, “IEEE 802.11e: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: MAC Enhance mentsfor Quality of Service (QoS),” Jan. 2005.

[16] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, “OAR: An Opportunistic Autorate Media Access Protocol for Ad Hoc Networks,” *SPRINGER Wireless Networks*, vol. 11, no. 1-2, pp. 39–53, 2005.

[17] J. Tourrilhes, “Packet Frame Grouping: Improving IP Multimedia Performance Over CSMA/CA,” *IEEE ICUPC 1998*, Florence, Italy, Oct. 5 – 9, 1998.

[18] V. Vitsas, P. Chatzimisios, A. Boucouvalas, P. Raptis, K. Paparrizos, and D. Kleftouris, “Enhancing Performance of the IEEE 802.11 Distributed Coordination Function

via Packet Bursting," *IEEE GLOBECOM Workshops 2004*, Dallas, Texas, USA, Nov. 29 – Dec. 3, 2004.

[19] Y. Lin and V. Wong, "Frame Aggregation and Optimal Frame Size Adaptation for IEEE 802.11n WLANs," *IEEE GLOBECOM 2006*, San Francisco, California, USA, Nov. 27 – Dec. 1, 2006.

[20] B. Ginzburg and A. Kesselman, "Performance Analysis of A-MPDU and A-MSDU Aggregation in IEEE 802.11n," *IEEE Sarnoff symposium 2007*, Princeton, New Jersey, USA, Apr. 30 – May 2, 2007.

[21] B. Kim, H. Hwang, and D. Sung, "Effect of Frame Aggregation on the Throughput Performance of IEEE 802.11n," *IEEE WCNC 2008*, Las Vegas, Nevada, USA, Mar. 31 – Apr. 3, 2008.

[22] X. He, F. Li, and J. Lin, "Link Adaptation with Combined Optimal Frame Size and Rate Selection in Error-prone 802.11n Networks," *IEEE ISWCS 2008*, Reykjavik, Iceland, Oct. 21 – 24, 2008.

[23] D. Skordoulis, Q. Ni, H. Chen, A. Stephens, C. Liu, and A. Jamalipour, "IEEE 802.11n MAC Frame Aggregation Mechanisms for Next-generation High-throughput WLANs," *IEEE Wireless Communications*, vol. 15, no. 1, pp. 40–47, 2008.

[24] C. Wang and H. Wei, "IEEE 802.11n MAC Enhancement and Performance Evaluation," *SPRINGER Mobile Networks and Applications*, vol. 14, no. 6, pp. 760–771, 2009.

[25] A. Riggi and J. Gomez, "RegionDCF: A self-adapting CSMA/Round-robin Media Access Protocol for WLAN," *IEEE LCN 2011*, Bonn, Germany, Oct. 4 – 7, 2011.

[26] Q. Liu, D. Zhao, and H. Ding, "An Improved Polling Scheme for PCF MAC Protocol," *IEEE WiCOM 2011*, Wuhan, China, Sep. 23 – 25, 2011.

[27] H.-W. Ferng, C. Setiadji, and A. Leonovich, "Fair Round Robin Binary Countdown to Achieve QoS Guarantee and Fairness in WLANs," *SPRINGER Wireless Networks*, vol. 17, no. 5, pp. 1259–1271, 2011.

[28] X. L. Huang and B. Bensaou, "On Max-min Fairness and Scheduling in Wireless Ad-hoc Networks: Analytical Framework and Implementation," *ACM MOBIHOC 2001*, Long Beach, California, USA, Oct. 4 – 5, 2001.

[29] S. Jain, S. R. Das, and H. Gupta, "Distributed Protocols for Scheduling and Rate Control to Achieve Max-Min Fairness in Wireless Mesh Networks," *IEEE WoWMoM 2007*, Helsinki, Finland, June 18 – 21, 2007.

[30] L. B. Jiang and S. C. Liew, "Proportional Fairness in Wireless LANs and Ad Hoc Networks," *IEEE WCNC 2005*, New Orleance, LA, USA, Mar. 13 – 17, 2005.

[31] B. Bensaou and Z. Fang, "A Fair MAC protocol for IEEE 802.11-based Ad Hoc Networks: Design and Implementation," *IEEE Transactions on Wireless Communications*, vol. 6, no. 8, pp. 2934–2941, 2007.

[32] J. He and H. K. Pung, "Fairness of Medium Access Control Protocols for Multi-hop Ad Hoc Wireless Networks," *ELSEVIER Computer Networks*, vol. 48, no. 6, pp. 867–890, 2005.

[33] T. Razafindralambo and I. Guérin-Lassous, "Increasing Fairness and Efficiency Using the MadMac Protocol in Ad Hoc Networks," *ELSEVIER Ad Hoc Networks*, vol. 6, no. 3, pp. 408–423, 2008.

[34] H. Luo, S. Lu, and V. Bharghavan, "A New Model for Packet Scheduling in Multihop Wireless Networks," *ACM MOBICOM 2000*, Boston, Massachusetts, USA, Aug. 6 – 11, 2000.

[35] B. Bensaou, Y. Wang, and C. C. Ko, "Fair Medium Access in 802.11 Based Wireless Ad-hoc Networks," *ACM MOBIHOC 2000*, Boston, Massachusetts, USA, Aug. 11, 2000.

[36] J. F. Lee, W. Liao, and M. C. Chen, "Interframe-Space (IFS)-Based Distributed Fair Queuing for Proportional Fairness in IEEE 802.11 WLANs," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 3, pp. 1366–1373, 2007.

[37] H. Luo and S. Lu, "A Topology-independent Wireless Fair Queueing Model in Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 3, pp. 585–597, 2005.

[38] Y. Kim and C.-H. Choi, "Analysis of the back-off mechanism and enhancing fairness in IEEE 802.11 ad-hoc networks," *Ph.D dissertation, Seoul National University*, 2009.

[39] D. Bertsekas and R. Gallager, *Data Networks (2nd ed.)*. Prentice-Hall, Inc., 1992.

[40] C. Chaudet, D. Dhoutaut, and I. Lassous, “Performance Issues with IEEE 802.11 in Ad Hoc Networking,” *IEEE Communications Magazine*, vol. 43, no. 7, pp. 110–116, 2005.

[41] Q. C. Douglas J. Leith and V. G. Subramanian, “Max-Min Fairness in 802.11 Mesh Networks,” *IEEE Transactions on Networking*, vol. 20, no. 3, pp. 756–769, 2012.

[42] A. Tsertou and D. Laurenson, “Revisiting the Hidden Terminal Problem in a CSMA/CA Wireless Network,” *IEEE Transactions on Mobile Computing*, vol. 7, no. 7, pp. 817–831.

[43] S. Razak, V. Kolar, and N. Abu-Ghazaleh, “Modeling and Analysis of Two-flow Interactions in Wireless Networks,” *ELSEVIER Ad Hoc Networks*, vol. 8, no. 6, pp. 564–581.

[44] Y. Kim and C.-H. Choi, “Analysis of the IEEE 802.11 Back-off Mechanism in Presence of Hidden Nodes,” *IEICE Transactions on Communications*, vol. 92, no. 4, pp. 1291–1299.

[45] Y. Kim, J. Yu, S. Choi, and K. Jang, “A novel hidden station detection mechanism in IEEE 802.11 WLAN,” *IEEE Communications Letters*, vol. 10, no. 8, pp. 608–610, Aug. 2006.

[46] K. Nishide, H. Kubo, R. Shinkuma, and T. Takahashi, “Detecting Hidden Terminal Problems in Densely Deployed Wireless Networks,” *IEEE GLOBECOM 2010*, Miami, Floorida, USA, Dec. 6 – 10, 2010.

[47] H. Ma, J. Zhu, and S. Roy, “On Loss Differentiation for CSMA-based Dense Wireless Network,” *IEEE Communications Letters*, vol. 11, no. 11, pp. 877–879.

[48] P. Karn, “MACA - A new channel access method for racket radio,” *ARRL/CRRL Amateur radio computer networking conference*, London, Ontrario Canada, Sep. 22, 1990.

[49] F. Talucci, M. Gerla, and L. Fratta, “MACA-BI (MACA by invitation)- A Receiver Oriented Access Protocol for Wireless Multihop Networks,” *IEEE PIMRC 1997*, Helsinki, Finland, Sep., 1997.

[50] R. Singh and D. Lobiyal, “Performance Modeling of Slotted MACA-BI MAC Protocol for Mobile Ad Hoc Networks,” *ACM ICIS 2009*, Seoul, Korea, Dec. 11, 2009.

[51] L. Du and L. Chen, “Receiver Initiated Network Allocation Vector Clearing Method in WLANs,” *IEEE Asia-Pacific Conference on Communications*, Oct. 2005.

[52] Z. Haas and J. Deng, "Dual Busy Tone Multiple Access (DBTMA)- A Multiple Access Control Scheme for Ad Hoc Networks," *IEEE Transactions on Communications*, vol. 50, no. 6, pp. 975–985.

[53] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance Anomaly of 802.11b," *IEEE INFOCOM 2003*, San Francisco, California, USA, Mar. 30 – Apr. 3, 2003.

[54] H. Kim, S. Yun, I. Kang, and S. Bahk, "Resolving 802.11 Performance Anomalies Through Qos Differentiation," *IEEE Communications Letters*, vol. 9, no. 7, pp. 655–657.

[55] Chou, C.T. and Shin, K.G. and Shankar, N. and others, "Contention-Based Airtime Usage Control in Multirate IEEE 802.11 Wireless LANs," *IEEE/ACM Transactions on Networking*, vol. 14, no. 6, pp. 1179–1192.

[56] A. Banchs, P. Serrano, and H. Oliver, "Proportional Fair Throughput Allocation in Multirate IEEE 802.11e Wireless LANs," *SPRINGER Wireless Networks*, vol. 13, no. 5, pp. 649–662.

[57] T. Joshi, A. Mukherjee, Y. Yoo, and D. Agrawal, "Airtime Fairness for IEEE 802.11 Multirate Metworks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 4, pp. 513–527.

[58] H. Lee and C.-H. Choi, "Achieving Airtime Fairness and Maximum Throughput in IEEE 802.11 under Various Transmission Durations," *IEICE Transactions on Communications*, vol. E94-B, no. 11, pp. 3098–3106.

[59] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, 2000.

[60] NS2 Network Simulator, http://www.isi.edu/nsnam/ns.

[61] Li, X. and Zeng, Q. A., "Capture Effect in the IEEE 802.11 WLANs with Rayleigh Fading, Shadowing, and Path Loss," *IEEE WiMob 2006*, Montreal, Canada, June 19 – 21, 2006.

[62] IEEE 802.11 WG, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," *IEEE Std.*, 2007.

APPENDICES

To provide the necessary materials for the development of the dissertation, this appendices are excerpted from the hidden node detection and resolving mechanisms in [8, 9] with only minor modification.

A. Hidden node detection mechanism

Kim *et al.* [8] proposed the hidden node detection mechanism using the new features of IEEE 802.11n, i.e., the frame aggregation and block ACK. Frame losses are categorized into two types: entire (E) and partial (P) frame losses during transmission according to a receiver's response. The receiver sends a block ACK back to the sender once the PHY header of a frame is decoded successfully, even when the subframes are not successfully received in IEEE 802.11n. Therefore, the frame loss is denoted as entire frame loss (E) when the sender cannot receives the block ACK from the receiver. Otherwise, it is denoted as partial frame loss (P). Then, frame losses can be subdivided according to their type and cause, as shown in Table A.1. For the entire frame loss type (E), frame losses can be caused by a collision (E1), hidden nodes (E2), and channel impairments (E3). Note that a collision always results in entire frame loss, not partial frame loss, because the frame transmissions that result in collision always have started at the same slot. Therefore, the partial frame losses are caused only by hidden nodes (P2) and channel impairments (P3). The sender can get more information on the transmission result from a block ACK, and this can help to differentiate frame losses according to their causes. Now, we describe

Table A.1 Classification of frame loss event.

Type of frame losses	Causes of frame losses	Receiver response
Entire frame loss (E)	(E1) Collisions (synchronous interference) (E2) Hidden nodes (asynchronous interference on PHY header) (E3) Channel impairments (during PHY header or block ACK frame transmission)	No response
Partial frame loss (P)	(P2) Hidden nodes (asynchronous interference on frame body[4]) (P3) Channel impairments (during frame body transmission)	Transmit block ACK

how to detect events (E2) and(P2), and a method for a sender to determine whether or not to use the RTS/CTS exchange.

Detecting entire frame losses caused by hidden nodes (E2)

A sender can estimate the probability of (E2) using measurable MAC layer statistics [8]. We consider an arbitrary node i with N_i transmitters within its carrier sensing range. Also, there may be hidden nodes out of the carrier sensing range of node i. Let us denote p_i^{ctr} as the probability that there is at least one node among N_i nodes concurrently transmitting with node i at the same slot. It is noted that not all of the concurrent transmissions by the nodes in the carrier sensing range result in a collision. The receiver can still decode a frame when the received SINR is larger than the SINR threshold, which is known as the capture effect [61], and we denote this probability as p_i^{cap}. Then, the probability p_i^{ack} that node i successfully receives a block ACK frame can be calculated as

$$p_i^{ack} \quad = p_i^{ctr} \cdot p_i^{cap} + (1 - p_i^{ctr}) \cdot (1 - p_{i,E}^{hid}), \tag{A.1}$$

where $p_{i,E}^{hid}$ is the probability of the PHY header becoming corrupted by interference from hidden nodes (E2). Let τ_i be the attempt probability of node i, and then p_i^{ctr} can be derived as

$$p_i^{ctr} = 1 - \prod_{j=1}^{N_i}(1 - q_{i,j}\tau_j), \qquad (A.2)$$

where $q_{i,j}$ is the conditional probability that node j senses the channel as idle given that node i senses the channel as idle. If the carrier sensing areas of nodes i and j are identical, the conditional probability $q_{i,j}$ then becomes 1. However, the carrier sensing areas can be different, and $q_{i,j}$ is less than 1 in general. The probability p_i^{idle} that no node initiates transmission within the carrier sensing range of node i in an idle slot can be expressed as

$$p_i^{idle} = (1 - \tau_i)\prod_{j=1}^{N_i}(1 - q_{i,j}\tau_j). \qquad (A.3)$$

Although it is difficult for node i to know the values of $q_{i,j}$ and N_i without exchanging information between the nodes in the carrier sensing range, $q_{i,j}$ and N_i can be eliminated by using the relationship between (A.2) and (A.3). That is, p_i^{ctr} can be simplified as

$$p_i^{ctr} = 1 - \frac{p_i^{idle}}{1 - \tau_i}. \qquad (A.4)$$

And, $p_{i,E}^{hid}$ in (A.1) can be expressed as

$$p_{i,E}^{hid} = 1 - \frac{p_i^{ack} - p_i^{ctr}p_i^{cap}}{1 - p_i^{ctr}}. \qquad (A.5)$$

Based on the relations between p_i^{ctr} in (A.4) and $p_{i,E}^{hid}$ in (A.5), we can estimate $p_{i,E}^{hid}$ by estimating p_i^{ack}, p_i^{cap}, τ_i, and p_i^{idle}. Note that the probabilities p_i^{ack}, p_i^{idle}, and τ_i can be estimated using MAC layer statistics. However, it is difficult to estimate p_i^{cap} because this value depends on the relative location of node i and the nodes that concurrently transmit. Thus, the lower bound $\underline{p}_{i,E}^{hid}$

for $p^{hid}_{i,E}$ is computed by setting $p^{cap}_i = 0$, and can be expressed as

$$\underline{p}^{hid}_{i,E} = 1 - \frac{p^{ack}_i (1 - \tau_i)}{p^{idle}_i}. \qquad (A.6)$$

After a sender estimates $\underline{p}^{hid}_{i,E}$ using MAC layer statistics, it calculates the averaged value of $\underline{p}^{hid}_{i,E}$ by low-pass filtering as follows:

$$\underline{p}^{hid}_{i,E} \leftarrow \rho \cdot \underline{p}^{hid}_{i,E} + (1 - \rho) \cdot \left[1 - \frac{p^{ack}_i (1 - \tau_i)}{p^{idle}_i} \right]. \qquad (A.7)$$

Once $\underline{p}^{hid}_{i,E}$ is estimated, it can be used to help the sender to make decision on whether to employ the RTS/CTS mechanism to resolve the hidden node problem. When $\underline{p}^{hid}_{i,E}$ is less than a certain threshold $\eta^{(E2)}$, the transmitter considers that an entire frame loss is caused mainly due to a collision. Otherwise, it comes to the conclusion that there are hidden nodes and moves to the next step to determine whether or not to use the RTS/CTS mechanism. Note that MAF-MAC can calculate $\underline{p}^{hid}_{i,E}$ in the same way, because it also uses the block ACK feature of IEEE 802.11n.

Detecting partial frame losses caused by hidden nodes (P2)

When a transmitter receives a block ACK frame, it only needs to differentiate partial frame losses due to hidden nodes (P2) from channel errors (P3), considering that a collision cannot occur in such a case. It is assumed that each sender transmits at an appropriate transmission rate in accordance with the channel condition by using feedback information of block ACK. Then, the hidden nodes are considered to be the main cause of the error if the error probability of subframes (MPDUs in [8] and fragments in MAF-MAC) is higher than expected. The sender can easily calculate the error probability for a subframe by simply counting the number of corrupted subframes in an entire frame. Let $p^{hid}_{i,P}$ be the error probability of a subframe due to interference from hidden nodes (P2). Upon receiving a block ACK, $p^{hid}_{i,P}$ is low-pass filtered with a coefficient ρ (0< ρ <1) as

$$p_{i,P}^{hid} \leftarrow \rho \cdot p_{i,P}^{hid} + (1-\rho) \cdot \frac{X}{Y}, \qquad (A.8)$$

where Y and X are the numbers of all subframes and corrupted subframes in an entire frame, respectively. The transmitter can detect the presence of hidden nodes when $p_{i,P}^{hid}$ is larger than a certain threshold $\eta^{(P2)}$. MAF-MAC can easily estimate $p_{i,P}^{hid}$ because it also uses the fragment aggregation and block ACK features.

When to employ RTS/CTS

When (E2) or (P2) is detected, the proposed HD mechanism [8] determines whether or not to initiate the RTS/CTS exchange depending on the estimated values of $p_{i,E}^{hid}$ and $p_{i,P}^{hid}$. It is better to initiate the RTS/CTS exchange when the RTS/CTS mode is expected to yield higher throughput than the basic access mode. In [8], the throughput THR of a node is derived based on the following simple throughput model that ignores the variations of τ_i and p_i^{ctr} by employing the RTS/CTS exchange.

$$THR = \frac{n_s \cdot S_p}{T_{data} + T_{oh}}, \qquad (A.9)$$

where n_s and S_p are the number of successfully transmitted subframes and the packet size, respectively. Note that T_{oh} is the overhead required for completing a transmission of a single data frame, which is calculated as

$$T_{oh} = \begin{cases} T_{SIFS} + T_{DIFS} + T_{BO}, & \text{for basic access mode,} \\ T_{rts} + T_{cts} + 3\,T_{SIFS} + T_{DIFS} + T_{BO}, & \text{for RTS/CTS mode,} \end{cases} \qquad (A.10)$$

where T_{rts}, T_{cts}, are the time durations for transmitting RTS and CTS frames, respectively. Then, the number of successfully transmitted subframes can be estimated as

$$n_s = \begin{cases} (1 - p_{i,E}^{hid})(1 - p_{i,P}^{hid})n_{ts}, & \text{for basic access mode,} \\ n_{ts}, & \text{for RTS/CTS mode,} \end{cases} \qquad (A.11)$$

where n_{ts} is the total number of subframes in a data frame. Let THR_i^B and THR_i^R be the throughput for the basic access mode and RTS/CTS mode, respectively. Node i calculates THR_i^B and THR_i^R based on the estimated $p_{i,E}^{hid}$ and $p_{i,P}^{hid}$ by using (A.9), (A.10), and (A.11), and employs the RTS/CTS exchange when THR_i^R/THR_i^B is larger than 1.

B. Hidden node resolving mechanism [9]

Kim [9] proposed two types of effective resolution mechanism for the hidden node problem; extending the effective CTS range and introducing the Receiver-oriented contention (ROC) mechanism.

Extending the effective CTS range

The RTS/CTS mechanism was designed to prevent the transmissions of the other nodes in the neighborhood of a transmitter or a receiver by announcing the information on the upcoming data transmission via RTS and CTS frames. In particular, the transmission of CTS frame is effective in preventing the transmission of hidden nodes, because they defer channel access by setting the NAV value appropriately when overhearing the CTS frame. In the IEEE 802.11 standard [62], the NAV value after receiving a CTS frame is calculated based on the value of the duration field in the CTS frame. In [9], the effective CTS range is defined as the range that the transmission of CTS frame can prevent the transmission of other nodes during the subsequent data and ACK transmissions. Then, the effective CTS range becomes equal to the transmission range of the CTS frame because only a node that can decode the CTS frame appropriately calculates the NAV value. If a node that is located out of the transmission range but within the carrier sensing range of the CTS frame, it cannot decode the CTS frame but only can sense the transmission of a frame. If the node defers its channel access by setting extended inter-frame space (EIFS) instead of NAV, there is no guarantee that the upcoming data transmission will be protected because the duration of EIFS is much shorter than that of NAV.

In extending the effective CTS range, there are two major problems that have to be resolved. One is how to identify the CTS frame when a node can only sense the transmission of a frame, and the other is how to set the NAV value to protect the upcoming data transmission. To identify the control frames, each control frame should have a unique size. Fortunately, in the IEEE 802.11n standard [1] the CTS and ACK frames can be differentiated by transmission duration because block ACK frame is much larger than that of CTS frame. After identifying the CTS frame, a node has to defer channel access to protect the upcoming data transmission, but it still does not know the appropriate duration of NAV. Kim [9] proposed to adjust the average size of a frame in a given time by using Frame size Adaptation (FA) scheme so that the average transmission time of data frame, T_{data} is close to a fixed reference time T_{ref}. Then, a node can defer the channel access appropriately by simply setting the duration of NAV as T_{ref} after identifying the CTS frame. In this way, the effective CTS range can be extended.

Receiver-oriented contention (ROC) mechanism

There is a limitation in applying the RTS/CTS mechanism to resolve the hidden node problem because the exchange of control packets also suffers from the hidden node problem caused by the carrier sensing mechanism at the sender [9]. To resolve the hidden node problem more effectively, the Receiver-Oriented Contention (ROC) mechanism, which is inspired by the basic

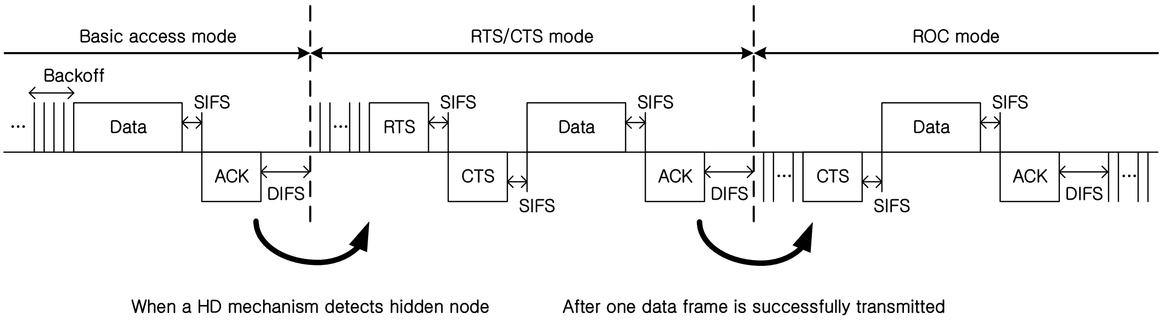


Figure A.1 Basic operation of the ROC mechanism in conjunction with the HD mechanism

idea of MACA-BI [49], is adopted. In MACA-BI, the part corresponding to the RTS frame transmission is suppressed and a receiver polls the transmitter for data transmission via RTR frame, which is a renamed CTS frame. The ROC mechanism can be simply incorporated in IEEE 802.11 system by allowing a receiver to participate in the contention to access the channel via the CSMA/CA mechanism. For this, a receiver has its own back-off counter, decrements the counter by one when the channel is idle, and transmits a CTS frame whenever the counter reaches zero. Upon receiving the CTS frame, the transmitter transmits a data frame to the receiver and subsequently the receiver transmits an ACK frame to the transmitter. The ROC mechanism can save the transmission time of RTS frames and protect the interference from hidden nodes because the receiver senses the channel and contends with its neighboring nodes.

However, there are some disadvantages in this mechanism. The most critical one is that different forms of hidden node problem may occur. Thus, the ROC mechanism is adopted with the hidden node detection (HD) mechanism, so that only the links that suffer from the hidden node problem use the ROC mechanism, as can be seen in Fig. A.1. Initially, the link operates in basic access mode and the transmitter detects the presence of a hidden node via the HD mechanism. When the interference from a hidden node is detected, the sender initiates the RTS/CTS mechanism to alleviate the interference from a hidden node. It is noted that the RTS/CTS exchange may still not be successful due to hidden nodes. In such a case, the transmission in the RTS/CTS mode needs to be continued until one data frame transmission is successful. After one successful transmission of a data frame in the RTS/CTS mode, the link switches to transmit in the ROC mode to resolve the hidden node problem more effectively and the receivers contend based on the Binary Exponential Backoff (BEB) mechanism as in the IEEE 802.11 DCF. MAF-MAC uses the ROC mechanism with HD to resolve the hidden node problem in section 4.3.

초록

최근 스마트폰, 태블릿 PC 등의 무선 네트워크를 사용하는 모바일 기기의 사용이 급증함에 따라 무선 랜 (wireless local area network (WLAN))에 대한 수요가 빠르게 증가하고 있다. 하지만, IEEE 802.11 표준에서 기본적으로 사용하는 MAC (medium access control) 프로토콜인 DCF (distributed coordination function) 는 single-cell 네트워크에서 MAC 효율 (MAC efficiency) 성능이 떨어지는 문제점과 ad-hoc 네트워크에서 노드간에 공평성 성능이 크게 저하 되는 문제점을 지니고 있다. 본 논문에서는 이러한 네트워크에서 DCF가 지니고 있는 문제점을 각각 해결할 수 있는 두 가지 다른 방식의 MAC 프로토콜들을 제안하였다. 기존의 MAC 프로토콜에서는 패킷 (packet) 이나 프레임 (frame) 의 크기가 정해지면, 각 노드 (node) 의 데이터 전송 속도에 따라 (data transmission rate) 프레임 전송 시간 (frame transmission duration) 이 정해졌다. 하지만, 본 논문에서는 IEEE 802.11n/ac/ad 표준에서 사용하는 프레임 결합 (frame aggregation) 과 block ACK 기법을 이용하여 프레임 전송 시간을 정확히 조절 할 수 있는 방법을 제안하였다. 만약 이와같이 프레임 전송 시간을 우리가 원하는 데로 정확하게 조절 할 수 있게된다면, 네트워크 상에 각 노드들은 추가적인 오버헤드 (overhead) 없이 자신이 알려주고자 하는 정보를 프레임 전송 시간을 이용하여 자신 주변의 노드들에게 간접적으로 알려줄 수 있게 된다. 즉, 프레임 전송 시간을 정확히 조절하는 것을 통해서 기존의 컨트롤 메시지 (control message) 가 수행했던 역할인 정보 전달의 역할을 수행 할 수 있게 된다. 이 아이디어는 간단하지만, 각 노드들이 네트워크 성능을 향상 시킬 수 있는 정보를 교환하는데 효과적이다. 본 논문에서 제안된 두 개의 MAC 프로토콜들은 이 아이디어를 활용하여 네트워크 성능을 향상시키고자 하였다.

우선, IEEE 802.11 single-cell 네트워크에서의 MAC 효율 성능을 향상시키기 위해 Transmission Order Deducing MAC (TOD-MAC) 프로토콜을 제안하였다. 최근 물리 계

층 (physical layer) 에서의 전송 속도가 Gbps 범위까지 비약적으로 발전하고 있다. 하지만, 이러한 물리 계층 전송 속도의 증가가 MAC 계층 (MAC layer) 에서의 처리량 (throughput) 성능 향상에 효과적으로 기여하지 못하고 있는 실정이다. 왜냐하면, 물리 계층에서의 전송 속도가 올라 갈수록 PHY header와 컨텐션 시간 (contention time) 등의 MAC 계층에서 발생하는 오버헤드들이 처리량 성능 향상에 큰 걸림돌이 되기 때문이다. 이러한 문제점을 해결 하기 위해서 TOD-MAC에서 각 노드들은 자신의 전송 순서에 따라 앞서 제안된 방법을 이용하여 프레임 전송 시간을 정확히 조절하여 데이터를 전송한다. 이를 통해 네트워크 상의 각 노드들은 자신 주변 노드들의 전송 순서를 프레임 전송 시간을 통해 추정할 수 있게 되고, 자신에게 알려진 전송 순서 정보를 이용하여 순환 순서 방식 (round robin manner) 으로 데이터를 전송한다. 이를 통해 제안된 MAC 프로토콜은 전송 충돌 (transmission collision) 과 컨텐션 시간을 효율적으로 줄일 수 있게 되고, CSMA/CA (carrier sensing multiple access with collision avoidance) 기반의 single-cell 네트워크에서의 MAC 효율을 극대화 시킬 수 있게 된다. 또한, 실험을 통해 TOD-MAC이 다양한 환경에서 높은 처리량 성능과, 좋은 short/long-term 채널 점유 시간 공평성 (air-time fairness) 성능을 보여주는 것을 확인할 수 있었다.

또한, 본 논문에서는 IEEE 802.11 ad-hoc 네트워크에서의 최대-최소 채널 점유 시간 공평성 (max-min air-time fairness) 을 향상 시킬 수 있는 Max-min Air-time Fairness MAC (MAF-MAC) 프로토콜을 제안하였다. 최근 IEEE 802.11 ad-hoc 네트워크를 기반으로한 서비스에 대한 요구가 빠르게 증하하면서, ad-hoc 네트워크에서 노드들 간에 공평한 서비스를 제공하는 것이 중요한 문제가 되고 있다. 이를 위해 MAF-MAC에서는 각 노드들이 자신의 채널 점유 시간에 대한 정보를 프레임 전송 시간을 통해 주변 노드들에게 알려주고, 각 노드들은 이 정보를 이용하여 자신의 contention window (CW) 값을 적절하게 조절하여 ad-hoc 네트워크에서의 최대-최소 채널 점유 시간 공평성 성능을 향상시키고자 하였다. 이를 통해 제안된 MAC 프로토콜은 네트워크에 있는 노드들에게 보다 공평한 서비스를 제공함과 동시에 채널 점유율과 사용율을 효율적으로 향상 시킬 수 있었다.

또한, 다른 연구에서 제안된 히든 노드 감지 (hidden node detection) 방법과 히든 노드 해결 (hidden node resolving) 방법을 MAF-MAC에 적용함으로써 ad-hoc 네트워크에서 발생할 수 있는 히든 노드 문제를 효과적으로 해결 할 수 있었다. 시뮬래이션을 통해 히든 노드의 존재 여부와 관계 없이 다양한 환경에서 MAF-MAC에 기반한 방법이 좋은 채널 점유 공평성 성능을 보여줌과 동시에 효율적으로 채널을 사용하고 있다는 것을 확인 할 수 있었다.

주요어 : 무선 랜, IEEE 802.11, IEEE 802.11n 표준, 다중 접근 제어, 순환 순서 방식, 전송 시간 조절, 최대-최소 공평성, 채널 점유 시간 공평성, 히든 노드 문제

학번 : 2010-30225

감사의 글

군대와 대학원 사이의 갈림길에서 많은 고민 끝에 대학원에 입학한지가 엊그제 같은데 벌써 5년 반이라는 시간이 흘러 저의 30을 맞이함과 동시에 졸업 하게 되었습니다. 대학원 시절동안 주변의 많은 분들의 도움으로 별 다른 어려움 없이 연구할 수 있었고 이를 통해 한단계 성장할 수 있는 좋은 기회였다고 생각합니다. 그동안 연구해 온 것을 정리하여 학위논문을 완성하게 되었습니다. 이 자리를 빌어 소중한 사람들에게 감사의 마음을 전합니다.

무엇보다 10년이 넘는 긴 대학 시절동안 부족함 없이 뒤에서 묵묵하게 지원해 주신 부모님에게 감사드립니다. 공부 안한다며 매일 걱정하시던 아버지, 철 없는 아들 공부시키느라 늦게 까지 일하시는 어머니 감사합니다. 자주는 볼 수 없지만 언제나 볼때마다 힘이 되어 주는 누나, 매형 고맙습니다. 그리고 볼때마다 웃음짓게 해준 귀여운 조카 서진이에게도 고맙다고 말을 하고 싶습니다.

여러모로 많이 부족한 제가 제어 및 시스템 연구실에서 공부하게 된것은 저에게 큰 행운이라 생각합니다. 우선 대학원에 입학할 때부터 석사과정을 거쳐 박사로 졸업할 때까지 부족한 저를 이끌어 주시고 많은 가르침을 주신 최종호 교수님 감사드립니다. 제가 항상 덜렁대서 미처 생각하지 못한 부분까지 세세하고 꼼꼼하게 지적해주셔서 논문을 작성하는데 많은 것을 배울 수 있었습니다. 또한, 청렴하신 인품과 항상 학생들을 걱정하는 교수님의 모습은 저에게 많은 것을 느끼게 해주었습니다. 앞으로 교수님께 배운

것들을 잊지 않고 성실하게 살도록 노력하겠습니다. 오랜 시간동안 옆에서 많은 것을 가르쳐주시고 대학원 생활을 즐겁게 할 수 있도록 도와준 제어 및 시스템 연구원분들께 감사의 말씀드립니다. 특히, 박사 과정동안 연구하는데 많은 도움을 주신 형호형, 민식이형, 민호에게 감사를 드립니다. 그리고 연구하는 분야는 다르지만 즐거운 연구실 생활을 할 수 있게 해준 상일형, 지용이형, 준희형, 정찬이형, 승호, 은우, 민규에게도 감사드립니다.

이제 모두들 할일이 생겨서 이전만큼 자주 보지는 못하지만, 만나면 언제나 즐거운 소중한 내 동네 친구들 진우, 강도, 재호, 이제는 멀리 홍성으로 내려간 유리, 전주에서 동물들 치료하는 안찬, 불러도 볼 수는 없는 영국에서 유학하는 원진, 얼마전 결혼한 노기 너희들이 있었기에 언제나 즐겁게 생활할 수 있었다. 그리고 고등학교 동창 백대, 윤상, 쭌맹, 고갱 등등 만날때마다 자극이 되고 즐겁게 놀 수 있어서 더욱 소중한 내 친구들 정말 고맙다. 대학 생활을 정말 즐겁게 해준 동문회 사람들 연락은 자주 못드리지만 정말 감사드립니다.

마지막으로 바쁘신 가운데 소중한 시간을 내 주셔서 논문을 심사해 주시고 많은 조언을 해주신 박세웅 교수님, 최성현 교수님, 오성회 교수님, 박은찬 교수님께도 큰 감사를 드립니다. 끝날것 같지 않았던 대학원 생활이 어느덧 끝났습니다. 오랜 시간동안 많은 것을 배울 수 있었지만, 아직은 많이 부족한 제가 이제 사회에 첫 발을 딛게 됩니다. 많은 분들에게 받은 도움과 사랑을 언제나 기억하며 새로운 마음가짐으로 성실하게 살아가도록 하겠습니다. 이 학위 논문을 저를 도와주신 많은 분들에게 바칩니다.