ECS: an Enhanced Carrier Sensing Mechanism for Wireless Ad-hoc Networks[1]

Zhifei Li
Department of Computer Science
Johns Hopkins University
Baltimore, MD 21218 USA
Email: zli19@jhu.edu

Sukumar Nandi
Dept. of Computer Science & Engineering
Indian Institute of Technology, Guwahati
India, 781039
Email: sukumar@iitg.ernet.in

Anil K. Gupta
School of Computer Engineering
Nanyang Technological University
Singapore, 639798
Email: asgupta@ntu.edu.sg

***Abstract*— In wireless ad-hoc networks, whenever a node overhears a frame, the node should defer its transmission to prevent the interference with the ongoing transmission. The exact duration value by which the node should defer is contained in the frame. However, due to the wireless transmission errors and due to the fact that the carrier sensing range is normally greater than the transmission range, a frame overheard by a node may not always be interpretable and thus the node cannot get the precise duration value by which the node should defer. Therefore, an important issue is that *whenever a node detects an erroneous frame, how long the node should defer for*. In the current IEEE 802.11 standards, the node will always defer the transmission by a fixed duration (represented by EIFS). We show that this duration is sometimes smaller and sometimes larger than the desired period by which the transmission should be deferred, and it leads to substantial unfairness and throughput degradation. We propose an enhanced carrier sensing (ECS) scheme, which distinguishes among the type of the erroneous frames based on their lengths and defers the transmission accordingly. Simulation results show that the ECS improves the fairness as well as the throughput substantially. To the best of our knowledge, this is the first work that focuses on the carrier sensing mechanism to improve the performance of IEEE 802.11.**

I. Introduction

Recently, wireless ad-hoc networks have attracted considerable research interest as they are easy to deploy and maintain. Since IEEE 802.11 [10] is the de facto standard for wireless LANs, most of the research work on wireless ad-hoc networks adopt it as the MAC layer. IEEE 802.11 defines two MAC protocols, i.e., Point Coordination Function (PCF) and Distributed Coordination Function (DCF). However, only DCF is used in wireless ad-hoc networks since the PCF requires centralized nodes (e.g., base stations). DCF is a CSMA/CA based protocol, which has three main components: Carrier Sensing (CS), Collision Avoidance (CA), and Contention Resolution (CR). In the DCF, the CA part includes the two-way and four-way handshaking, which aim to decrease the chances of collisions among the transmissions of the hidden terminals. On the other hand, the CR part is the well-known Binary Exponential Backoff (BEB) algorithm, which, based on the congestion status on the shared medium, dynamically adjusts the probability with which a node should access the medium. Since CS, CA, and CR all operate in a distributed manner without having precise information of the contention on the shared medium, they result in substantial unfairness. In addition, the distributed nature of IEEE 802.11 also leads to throughput degradation. In particular, sometimes the medium is unnecessarily idle while at other times the medium experiences collisions, both of which lead to bandwidth wastage.

In the literature, there has been a lot of research-work on improving fairness or throughput by modifying the CA or CR mechanism. Specifically, [2] mainly focuses on the CA part to improve the fairness, while [7], [16], and [17] aim to achieve fairness through modifying the CR algorithm of IEEE 802.11. In contrast, [1], [8], and [12] aim to enhance the CA handshake to improve the throughput by preventing collisions as much as possible, while [3] and [4] aim to make the CR algorithm more adaptive and thus improve the capacity utilization. It is necessary to point out that each of the above work either improves the fairness at the cost of the throughput, or the vice versa. Moreover, to the best of our knowledge, there is no research-work focusing on the carrier sensing (CS) part to improve the performance of the MAC protocol in ad-hoc networks. In this paper, we aim to improve the performance (both fairness and throughput) by enhancing the CS mechanism in IEEE 802.11.

In an IEEE 802.11-based wireless ad-hoc networks, since all the nodes share a common medium, only one flow among the contending flows can transmit during a certain duration. Therefore, whenever a frame exchange sequence between two nodes is in progress, it is extremely important to guarantee that all the other nodes in the interference range should defer their own transmission. In IEEE 802.11, whenever a node detects a physical carrier on the medium, it does not transmit. This is known as the *physical carrier sensing*. Moreover, even after the medium becomes idle, the node may need to defer further to allow the transmission of the *remaining* frames in the same sequence. This is known as the *virtual carrier sensing* (VCS). Under VCS, every frame will carry a duration value indicating the time by which the overhearing nodes should defer. Clearly, whenever a node overhears a frame, the node must be able to interpret the contents of the frame to get the exact length of the duration by which it should defer. However, this cannot be achieved in the following two cases. *(i)* If a frame experiences

[1]A concise version of this paper has been accepted for publication in the Proceedings of IFIP Networking 2004, Athens, Greece, May 9-14, 2004.

a transmission error, a node that overhears this frame cannot interpret the contents of the frame. *(ii)* Since the carrier sensing range is normally greater than the transmission range [18], whenever a frame is transmitted over the medium, a node out of the transmission range but within the sensing range of this transmission will detect the frame but it cannot interpret the contents. The above two cases occur very frequently in wireless networks and have a great impact on the performance of IEEE 802.11. Therefore, an important issue is that *whenever a node detects an erroneous frame, how long should the node defer its transmission*? To the best of our knowledge, this issue has not been addressed in the literature.

In the current IEEE 802.11 standards, whenever a node detects an erroneous frame on the medium, it defers the transmission by a *fixed* duration indicated by the Extended Inter-Frame Space (EIFS) constant. In this paper, we show that in some situations, the EIFS value is too large compared to the desired value, while in some other situations the EIFS value is too small. Respectively, we refer to these two cases as *(i)* large-EIFS problem, and *(ii)* small-EIFS problem. The two problems, together, are referred as the imprecise-EIFS problem. The imprecise-EIFS problem leads to immense unfairness and throughput degradation. Specifically, when the small-EIFS problem occurs, a node may begin its transmission even though some other nodes in the interference range are still transmitting, resulting in collisions (and thus throughput degradation). On the other hand, when the large-EIFS problem occurs, the medium may be unnecessarily idle and the node may experience unfairness as it defer a longer duration than other nodes. In order to solve the imprecise-EIFS problem, we propose an enhanced carrier sensing (ECS) mechanism in which the EIFS value is made variable in an adaptive manner. The simulation results show that our ECS greatly improves the fairness as well as the capacity utilization.

The rest of the paper is organized as follows. In Section II, the small- and large-EIFS problems are illustrated with the help of simple examples. The enhanced carrier sensing (ECS) mechanism is proposed in Section III. The performance of ECS is studied in Section IV. Section V concludes the paper.

II. Imprecise-EIFS Problem in IEEE 802.11

A. Preliminaries

A special characteristic of wireless propagation is the attenuation of the transmission power over the distance traversed by the signal. Based on the attenuation, two ranges are defined: the transmission range (TR) and the sensing range (SR). Normally, the SR range is greater than the TR range [18]. Correspondingly, we call a frame detected by a node within TR as the *TR frame*, and a frame detected by a node out of TR but within SR as the *SR frame*. If there is no transmission error, a TR frame can be received correctly. Clearly, if a TR frame experiences a transmission error, its contents are not interpretable. In contrast to a TR frame, an SR frame can be detected by the carrier sensing but it cannot be received correctly, and therefore will be treated as an error. Though we have mentioned two types of *erroneous* frame (i.e., a TR frame experiencing a transmission error, or an SR frame), we do not explicitly consider the case of a TR frame with a transmission error since in this case similar problems are resulted as in the case of an SR frame.

To cope with the hidden-terminal problem, IEEE 802.11 defines a four-way handshaking, where a sequence of Request To Send (RTS), Clear To Send (CTS), Data, and Acknowledgement (ACK) frames, is transmitted for the transmission of every single data packet. For the convenience, we call the exchange of RTS/CTS/Data/ACK frames as a frame exchange sequence (FES). $FES(X, Y)$ represents a FES between nodes X and Y, initiated by node X.

We now describe the simulation environment. NS-2 with CMU wireless extensions [6] is used for the simulations. In the simulation, each packet is 1000-bytes long and the raw bandwidth of the shared medium is set with 2 Mbps, leading to a maximum throughput about 1.4 Mbps due to the overhead in IEEE 802.11. For each flow, a Constant Bit Rate (CBR) traffic is adopted and the traffic source rate is made large enough for the single flow to occupy the entire channel capacity as unfairness occurs only when the system is over-loaded. Mobility is not considered and static routing is used. The above assumptions are adopted to clearly bring out the problems caused by the carrier sensing mechanism. The transmission range and sensing range are 250 and 550 meters, respectively. Other system parameters are set according to the Direct Sequence Spread Spectrum (DSSS) [10], i.e., SlotTime: 20 μs; SIFS: 10 μs; DIFS: 50 μs; EIFS: 364 μs.

B. Large-EIFS Problem

To explain the large-EIFS problem, scenario shown in Figure 1 is used, where the distance between two neighboring nodes is 200 meters. Therefore, nodes A and C out of TR but within the SR of each other. There are two single-hop flows, and we expect them to share the bandwidth equally, i.e., each flow should have a throughput of about 0.7 Mbps. However, from the results in Figure 2, we find that the flow from B to C gets about 1.15 Mbps whereas the flow from A to B gets only about 0.25 Mbps, which shows how much unfair the IEEE 802.11 is in this simple scenario.

Now we explain the reason of the above unfairness. IEEE 802.11 defines how a node should defer its transmission while a FES between two other nodes is in progress. Figure 3(a) demonstrates the process how node C defers its transmission while $FES(A, B)$ is in progress. First, node A sends a RTS to B. Since node C is out of the TR but within the SR of node A, it gets an SR frame. After the completion of this frame, node C defers its transmission with EIFS. Then, node B sends a CTS to node A. Since node C is within the TR of node B, it gets a TR frame and updates the Network Allocation Vector (NAV) with the duration contained in the CTS frame whose value is equal to *SIFS + TxtTime (Data) + SIFS + TxtTime (ACK)*. After this, node A sends the Data frame to node B. After the completion of this Data frame, node C defers its transmission with EIFS. At last, node B sends an ACK to node A. Since the duration field in the ACK frame is zero, the NAV of C will be updated to zero. At this point, all the three nodes will defer their transmission by DIFS, and then begin to contend for the medium using the random back-off. In the above scenario, all the nodes get up-to-date information of the medium, avoiding collision and guaranteeing fair access.

Now let us consider another scenario shown in Figure 3(b), in which node A is deferring while $FES(B,C)$ is going on. Since node A receives the TR frames corresponding to RTS and Data frames transmitted by node B, node A gets the precise state information of the medium up to the point when node C sends an ACK to node B[2]. Corresponding to this ACK, since node A is in the sensing range of node C, node A gets an SR frame and defers its transmission with EIFS. We can see that, after the completion of the ACK, before node A backs-off by a random period, it has to defer by EIFS duration rather than by DIFS as done by the other nodes (e.g., nodes B and C). Since EIFS is equal to *SIFS + TxTime (ACK) + DIFS* [10], the deferment at node A is certainly of much longer duration than that at the other nodes. We call this large-EIFS problem because the EIFS value is larger than it should be to reflect the state of the medium. In fact, the EIFS value should be equal to DIFS in this case.

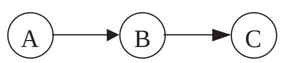

Fig. 1. 3-nodes with Two Single-hop Flows (Scenario-1)

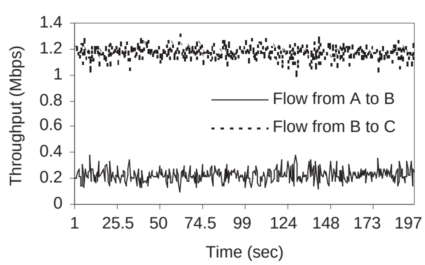

Fig. 2. Throughput under Large-EIFS Problem

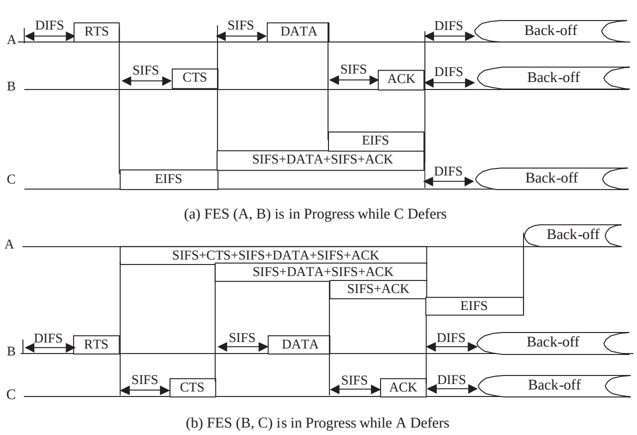

Fig. 3. Time Diagram Showing Frame Exchange Sequence

[2]Note that in Figure 3(b) we do not indicate the EIFS deferment at node A when it detects the SR frame corresponding to node C's CTS because the remaining NAV value is greater than the EIFS.

Summarizing the results of the above two cases, we see that whenever a $FES(A,B)$ is successfully completed, nodes A and B start contending for the medium at about the same time. However, whenever a $FES(B,C)$ is completed, node A contends for the medium *(EIFS−DIFS)* duration *later* than node B does, which is clearly very unfair for A. This explains the unfairness between the two single-hop flows. However, the above arguments do not explain why the throughput between the two flows differs so much. The average throughputs can be obtained by computing the relative frequencies of $FES(A,B)$ and $FES(B,C)$. In the Appendix, we present a Markov-chain model to obtain the average throughputs of the two flows, validating the results shown in Figure 2.

C. Small-EIFS Problem

While the large-EIFS problem causes substantial unfairness as shown in the previous subsection, here we will show that the small-EIFS problem results in substantial throughput degrade. The scenario of Figure 4 is used, where the distance between two neighboring nodes is again 200 meters. The simulation results show that the two flows share the bandwidth equally (in a long term). However, each of them only gets about 0.31 Mbps, resulting in an aggregate throughput about 0.62 Mbps, which is much smaller than the capacity (i.e., 1.4 Mbps).

To explain why the throughput degrades so much, let us consider the situation where nodes C and D are deferring their transmission, while $FES(A,B)$ is in progress (see Figure 5). After node B sends a CTS to A, node D gets an SR frame and defers its transmission with EIFS. Then, node A begins to send its Data frame. However, since node D is out of the SR of A, node D cannot hear this Data frame and thus it determines that the medium be idle though in fact it is busy. Moreover, since the EIFS value is much smaller than the transmission time of the Data frame, after deferring for the EIFS duration, node D backs-off and then sends out a frame even though node A is still transmitting the Data frame. The two transmissions result in a collision at node B and thus node B discards the frames. Likewise, due to symmetry in the topology, node C is also likely to drop the Data frame of a $FES(D,C)$, explaining why the throughputs degrade so much. We call this the small-EIFS problem because the EIFS value is smaller than it should be to reflect the state of the medium. In fact, in this situation, the EIFS value should be large enough to allow for the complete transmission of the Data frame. Note that in this scenario, as indicated in Figure 5, the large-EIFS problem also occurs at node D after node B sends the ACK to node A.

Fig. 4. 4-nodes with Two Single-hop Flows (Scenario-2)

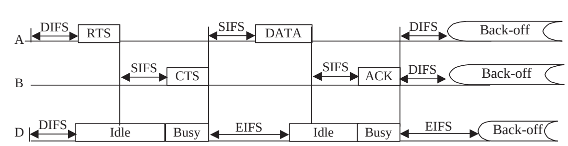

Fig. 5. Time Diagram when $FES(A,B)$ is in Progress while D defers

In the above discussion, we assumed that the capture is not supported. When the capture is supported, node B *may* capture (i.e., receive correctly) the Data frame from node A, even when there is interference from node D. Therefore, the throughputs of the flows will be greatly improved. However, we should resolve the small-EIFS problem irrespective of whether or not the capture is supported, due to the flowing three reasons: *(i)* Node B can capture the Data frame *only if* the Signal to Noise Ratio (SNR) of that frame is higher than a given threshold. This may not always be possible as the SNR depends on the relative distance of the nodes. Therefore, supporting capture cannot entirely make up for the throughput degradation caused by the small-EIFS problem; *(ii)* To support the capture ability, a node needs special functionality (e.g., high sensitivity) at the physical layer; and *(iii)* When node B *does* capture the Data frame successfully, the small-EIFS problem leads to unfairness as explained below. When node A is transmitting its Data frame, due to the small-EIFS problem, node D may initiate a frame, which will definitely be dropped since the frame sent by node D arrives at node C later than node A's Data frame. Then, node D will increase its CW and back off. Since the transmission time of the Data frame is very large compared to the back-off duration at node D, node D will attempt to send out a frame several more times during the node A's Data frame transmission. Every time D's frame is dropped, its Contention Window (CW) increases, leading to a large CW. In contrast, node A's Data frame can be captured by node B, and thus node A does not increase its CW. Clearly, this is unfair for node D and will result in substantial short-term unfairness between nodes A and D. The short-term unfairness may result in the stalling of on-demand routing protocols (e.g., AODV) as shown in [14]. In summary, when the capture *does* happen, the transmission from node D (due to the small-EIFS problem) *may* not affect the transmission of node A. However, it will affect the node D itself and result in short-term unfairness.

Another thing to note is that in the simulation, the static routing is used. If the dynamic routing protocols (e.g., AODV and DSR) are used, the flow that starts first (e.g., the flow from A) will get the entire bandwidth while the other flow (i.e., the flow from D) will be completely starved since the sender of the flow starting *later* cannot even find a route to the destination due to the imprecise-EIFS problem. This again shows the importance of solving the imprecise-EIFS problem. In [14], by varying the factors including the capture ability, with and without imprecise-EIFS problem, and the routing protocols, we have conducted an extensive simulation study for this topology. Our study shows that all the three factors greatly affect the performance. However, as our focus here is to study the effects of imprecise-EIFS problem, we assume that the capture is not supported and the static routing is used.

D. General Scenario involving imprecise-EIFS Problem

So far using specific examples, we have demonstrated how the large-EIFS and small-EIFS problems result in substantial unfairness and throughput degrade. In fact, which specific type of problem that a node will suffer from depends upon what kind of frame it detects. What kind of frame a node detects during an ongoing FES, in turn, depends upon the location of the node with respect to the location of the two nodes between whom the FES is in progress. For example, if a node gets an SR frame corresponding to a CTS, it may suffer from the small-EIFS problem. If a node gets an SR frame corresponding to an ACK, it may suffer from the large-EIFS problem. However, if a node gets an SR frame corresponding to a Data frame, the large- or small-EIFS problem does not arise because the EIFS value caters for the time needed for the next frame (i.e., ACK) to pass through. This is also true when an SR frame corresponding to a RTS is detected, since the next frame after a RTS is a CTS frame, and the length of the CTS is equal to an ACK. Figure 6 shows a generalized scenario with imprecise-EIFS problem where $FES(A, B)$ is in progress. We classify the entire region into seven areas. Nodes within different areas get different type of frames and thus may suffer from different type of problems. The results are summarized in Figure 7, where T(frame) denotes a TR frame and S(frame) denotes an SR frame. Note that in the area 6, a node will not suffer from the small-EIFS problem after detecting an SR frame of CTS, since the node can also detect the SR frame corresponding to the Data.

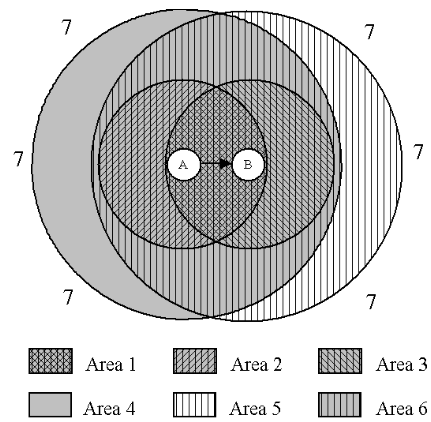

Fig. 6. Areas of Influence Due to $FES(A, B)$

Area 1: T(RTS) ->T(CTS) ->T(Data) ->T(ACK)
no large- or small-EIFS problem

Area 2: T(RTS) ->S(CTS) ->T(Data) ->S(ACK)
large-EIFS problem after S(ACK)

Area 3: S(RTS) ->T(CTS) ->S(Data) ->T(ACK)
no large- or small-EIFS problem

Area 4: S(RTS) ->S(Data)
no large- or small-EIFS problem

Area 5: S(CTS) ->S(ACK)
small-EIFS problem after S(CTS)
and large-EIFS problem after S(ACK)

Area 6: S(RTS) ->S(CTS) ->S(Data) ->S(ACK)
large-EIFS problem after S(ACK)

Area 7: Cannot detect any TR or SR frame
no large- or small-EIFS problem

Fig. 7. Problems Arising in Different Areas while $FES(A, B)$ is in Progress

III. Enhanced Carrier Sensing (ECS)

The basic idea behind EIFS deferment is as follows. When a node detects an SR frame, it assumes that this frame should be one of the frames belonging to an ongoing frame exchange sequence between two *other* nodes, and therefore, there may be another frame transmission in the near future between the same pair of nodes. In order to avoid interfering with the ongoing exchange, the node that detects the frame should defer its transmission by a certain duration. Since the *next* frame in the ongoing frame exchange sequence can be of any type (i.e., CTS, Data, or ACK), the transmission time required by this next frame may differ substantially. However, the IEEE 802.11 does not distinguish among different SR frames and uses the same constant EIFS value in all the cases, which results in large- and small-EIFS problems. Therefore, if somehow a node can distinguish among different type of SR frames and adopts different value of EIFS accordingly, these two problems can be greatly reduced and thus the performance (i.e., fairness and throughput) can be improved. In the light of this discussion, we make two proposals as follows:

1. Whenever a node detects an SR frame, it should try to identify the type of that frame.

2. EIFS value should be directly linked to the type of the SR frame detected.

Since our CS mechanism tries to identify the type of an SR frame rather than just sensing its presence, we call it Enhanced Carrier Sensing (ECS). In subsection A we discuss a scheme, which can be used to distinguish among the frames. Once the type of the frame is detected, how to choose the corresponding EIFS value is discussed in subsection B.

A. Distinguishing among SR Frames

Basically, there are two methods in which we can distinguish among different type of SR frames. The first method is to use the recent history of frames observed on the medium to decide the type of the current SR frame as the frames are transmitted in a specific order. However, this method is very complex and the recent history may not provide very precise information as there may be multiple FESs in progress at any given time. The second method is to get the information from the frame itself. However, it is not trivial as the node cannot interpret the contents of an SR frame. We propose that the various type of frames should have different lengths (in terms of bytes), and based on the length of an SR frame observed on the medium, the type of the frame is identified.

We first need to differentiate the lengths of different type of frames. In IEEE 802.11 [10], the lengths corresponding to the control frames are as follows, RTS: 20 bytes, CTS: 14 bytes, ACK: 14 bytes. On the other hand, the header of a Data frame is 34 bytes, implying that the length of a Data frame must be greater than 34 bytes. In order to distinguish between CTS and ACK, the size of CTS should be increased by a few bytes. The reason why we increase the size of CTS rather than ACK is that the extra bytes in the CTS may be used by the receiver to add some receiver status information, which may be useful to the sender before it sends out the Data frame. What is an appropriate number of bytes that should be added to the CTS depends upon the trade-off between the sensitivity of the physical layer and the additional overhead introduced due to these bytes. In our implementation, we change the length of the CTS to 17 bytes.

When a node gets an SR frame, to identify the type of the frame, the node only needs to detect the length (in terms of bytes) of that frame. Before discussing how to detect the length, we have to introduce the format of the physical layer frame in the IEEE 802.11 [10]. As presented in Figure 8, a Physical Layer Convergence Protocol (PLCP) frame includes three parts: PLCP Preamble, PLCP Header, and MAC Protocol Data Unit (PDU). The PLCP preamble is used by the receiver to synchronize, while the PLCP header contains information to help the receiver to decide the end of the frame. The MAC PDU corresponds to the MAC layer frame, e.g., a RTS frame.

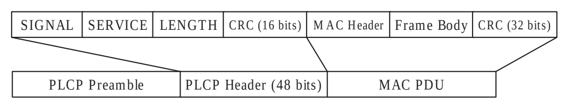

Fig. 8. PLCP Frame Format

Now we discuss how to detect the length (in terms of bytes at the MAC layer) of an SR frame by considering two possibilities. The *first* possibility is that the PLCP header part of the frame can be correctly received by the physical layer but the MAC PDU part cannot be interpreted by the MAC layer. This is very likely due to the following two reasons. *(i)* In the IEEE 802.11, the PLCP header is always transmitted at the lowest transmission rate (i.e., 1 Mbps) while the MAC PDU in the same PLCP frame may be transmitted at a higher rate (e.g., 2 Mbps or even higher). Generally, the Bit Error Rate (BER) under the case of lower transmission rate should be smaller if the Signal to Noise Ratio (SNR) is assumed to be the same, which should be true during the transmission of an entire PLCP frame. Therefore, the BER for the PLCP header may be much smaller than that for the MAC PDU. *(ii)* Moreover, the length of the PLCP header (i.e., 6 bytes) is much smaller than that of the MAC PDU (e.g., 20 bytes for an RTS frame). Due to the above two reasons, it is very likely that the PLCP header is correctly received while the MAC PDU is not. In such a situation, the MAC frame length (and thus the type) can be easily identified. Specifically, when the physical layer detects such a frame, it calculates the length of the MAC PDU based on the following two fields contained in the PLCP header: LENGTH field and SIGNAL field. The LENGTH field indicates the transmission time (in terms of μs) of the MAC PDU while the SIGNAL field indicates the transmission rate (in terms of 100 kbit/s) for the MAC PDU. Therefore, the length (in terms of bytes) of the MAC PDU can be detected if the PLCP header is received correctly. It is necessary to point out that the above discussion also applies for a TR frame containing transmission errors since the PLCP header of a TR frame should always be interpretable.

The *second* possibility is that, when an SR frame is detected, even the PLCP header part in the PLCP frame cannot be correctly received by the physical layer. In such a situation, to identify the type of the MAC frame, we can make use of the Clear Channel Assessment (CCA) mechanism [10]. Specifically, based on the length of the *time duration* that the CCA mechanism indicates a busy medium, if the transmission rate is fixed for all the frames, the MAC layer can calculate the length (in terms of bytes) and thus identify the type. However, when the transmission rate of the MAC PDUs varies, it may not work since two frames of different lengths (in terms of bytes), may have the same transmission time. Therefore, we should make the transmission time needed by various type of frames at different rates unique. It is easy to verify that the control frames among themselves satisfy this requirement if the length of the CTS frame has been changed as discussed above. However, a Data frame of a small length may be confused with the control frames. Therefore, whenever a data frame may be confused with a control frame, we should append enough dummy bytes in the Date frame, so that the transmission time needed by the Data frame is different from that of any control frame at any standard rate. Since there are only a few transmission rates (e.g., four rates in IEEE 802.11b) standardized, the method described above should be easily implemented. Therefore, even in the situation that the PLCP header is not received correctly and that the transmission rate is variable, the type of an SR frame can also be identified with the help of the CCA mechanism. However, for the simplicity, in our simulation the transmission rate of the MAC PDU is fixed. Also, the length of the Data frame is long enough (e.g, 1000 bytes) to avoid the confusion between a Data frame and a control frame.

Further comments: Whenever a node detects an SR frame, whether or not the PLCP header of that frame can be received correctly depends on how large the carrier sensing energy threshold [10] is. If the threshold is large enough, the PLCP header can always be interpreted whenever an SR frame is detected. What is the optimal value of the the carrier sensing threshold is out of the scope of this paper. However, according to the current standards [10], the case that the PLCP header is interpretable (i.e., the *first* possibility discussed before) is more likely to happen than the *second* possibility.

Also note that, while our ECS will get help from the physical layer to detect the type of a frame, it does not need any additional cooperation (compared to the current standards) between the MAC and physical layers, because the length of a frame (needed under the *first* possibility) and the CCA busy time (needed under the *second* possibility) are already conveyed to the MAC layer [10]. Therefore, the only change needed is to add some dummy bytes in the CTS frame, and sometimes in the Data frame if multi-rate is used under the *second* possibility as discussed before. Consequently, ECS can be easily incorporated in the IEEE 802.11.

B. EIFS Values in ECS

Based on the type of the frame observed on the medium, an EIFS value should be chosen accordingly. The basic rule for deciding the EIFS value is that it should be large enough to allow the complete transmission of the next frame in the sequence. Based on this rule, five different values of EIFS are defined in Figure 9. For a RTS type SR frame, the EIFS value is equal to SIFS + TxTime (CTS). When a node detects a CTS type SR frame, since the next frame in the sequence is a Data frame whose length may be variable and cannot be detected from the CTS type SR frame, we simply set the EIFS according to the maximum length allowed for the Data frames, i.e. Max-Data-Length. For a Data type SR frame, the EIFS value is set to SIFS + TxTime (ACK). Lastly, for an ACK type SR frame, since this is the last frame in the sequence, the EIFS value is set to DIFS.

In the discussion of the *second* possibility in Section III.A, we have deliberately ignored two complex problems for convenience of description. The first problem occurs as follows. In the wireless ad-hoc networks, due to the mobility and dynamic propagation characteristics (e.g., fading), when an SR frame is detected by a node, it may only be a part of a frame rather than being a complete frame. We call this partial-frame problem. In such a situation, we may not identify the type of the frame or misunderstand the type of the frame. Another type of problem may occur as follows. Because of the multi-hop scenario in the ad-hoc networks, spatial reuse of the bandwidth is very likely. Therefore, multiple frames may be there around a node at any given instant, resulting in a collision. As a result, the SR frame that a node detects may be an overlapped of several frames rather than a single frame. We call this overlapping-frames problem. In fact, it is very difficult for an MAC protocol to fully take care of these two problems. Therefore, whenever any of these two problems occur, we use the same value of EIFS as defined in the standard [10] . This is indicated by the last line in Figure 9. Note that the above two problems does not exist if the PLCP header of an SR frame can be interpreted, which is more likely as discussed before.

The ECS proposed above applies to both the four-way handshaking and the two-way handshaking defined in IEEE 802.11, though the four-way handshaking is more desirable for a multi-hop scenario.

```
EIFS(RTS) = SIFS + TxTime(CTS);
EIFS(CTS) = SIFS + TxTime(Max-Data-Length);
EIFS(Data) = SIFS + TxTime(ACK);
EIFS(ACK) = DIFS;
EIFS(collision or unknown type) = Standard EIFS;
```

Fig. 9. Different EIFS Values in ECS

IV. Simulation Results

In the performance evaluation of our Enhanced Carrier Sensing (ECS), the same simulation parameters are used as described in Section II. In Section IV.A, the performance is studied for the simple scenarios having only two flows. Then, in Section IV.B, two complex scenarios (i.e., chain and double-ring topologies) are studied.

A. Performance for Scenarios with Two-flows

Scenario-1: This scenario exhibited in Figure 1 has been used to demonstrate the large-EIFS problem in Section II. The average throughputs under the scenario are presented in Table I. The IEEE 802.11 is very unfair due to the large-EIFS problem at node A. However, under our ECS, the two flows share the bandwidth equally, i.e., each obtains an average throughput of about 0.7 Mbps, and thus we conclude that the large-EIFS problem has been eliminated up to a great extent. Moreover, as shown in Table I, the aggregate throughput under the ECS is slightly greater than that under IEEE 802.11. This can be explained as follows. Consider that after a successful transmission by node B, nodes A and B contend for the medium. Suppose that the node A generates a random back-off timer equal to 10 slots, while the node B generates a random back-off timer with 30 slots, the node A will get control of the medium. If the large-EIFS problem does not occur as in the case of ECS, the medium will be idle for only 10 slots and then node A begins to transmit. On the contrary, if the large-EIFS problem occurs as in the case of IEEE 802.11, the medium will be idle by an extra duration equal to $(EIFS - DIFS)$, i.e., 16 slots. In summary, since the ECS solves the large-EIFS problem, it reduces the idle time of the medium, explaining the throughput improvement in ECS. Though the absolute throughput improvement is not much (about 15 Kbps) in the given scenario, it should be much higher if a high-rate physical layer is used.

TABLE I

THROUGHPUT COMPARISON FOR SCENARIO-1

Throughput (Mbps)	IEEE 802.11	ECS
A to B	0.254	0.705
B to C	1.154	0.718
Aggregate	1.408	1.423

Scenario-2: This scenario (Figure 4) has been used to demonstrate the small-EIFS problem in Section II. The average throughputs are presented in Table II. Under the IEEE 802.11, due to the small-EIFS problem, the aggregate throughput is very small (i.e., 0.621 Mbps). However, under our ECS, the aggregate throughput greatly improves (i.e., 1.334 Mbps) as the ECS solves the small-EIFS problem. Note that our ECS does not require capture ability at the physical layer. Also, note that the static routing is used in the simulation. In [14] we have done a detailed study for this topology where the capture is supported and the dynamic routing is used.

TABLE II

THROUGHPUT COMPARISON FOR SCENARIO-2

Throughput (Mbps)	IEEE 802.11	ECS
A to B	0.314	0.662
D to C	0.307	0.672
Aggregate	0.621	1.334

Scenario-3: Now we discuss the results for the scenario shown in Figure 10. This scenario is similar to scenario-2 except that the direction of the flows is reversed. It is easy to see that after the node A sends back an ACK frame to node B, node C suffers from the large-EIFS problem if IEEE 802.11 is used. This is also true for node B after the node D sends back an ACK frame to node C. The average throughputs of the two flows are presented in Table III. In contrast to scenario-1, the large-EIFS problem does not result in long-term unfairness in this scenario (i.e., otherwise the flows will have different average throughputs). However, it results in short-term unfairness as shown later.

Table III also shows that the aggregate throughput under the ECS improves marginally in comparison to the IEEE 802.11. This is because the ECS solves the large-EIFS problem and thus reduces the chance of the medium being idle, as explained in the discussion of scenario-1.

Fig. 10. 4-nodes with Two Single-hop Flows (Scenario-3)

TABLE III

THROUGHPUT COMPARISON FOR SCENARIO-3

Throughput (Mbps)	IEEE 802.11	ECS
B to A	0.708	0.719
C to D	0.702	0.710
Aggregate	1.410	1.429

In order to study fairness on a short-term basis, we need to make use of the popular Jain's fairness index [5], which is defined as follows:

$$F_J = (\sum_{i=1}^{N} \gamma_i)^2 / (N \sum_{i=1}^{N} \gamma_i^2)$$

where *N* is the total number of flows who share the wireless medium, and γ_i is the fraction of the bandwidth utilized by flow *i* over a certain number of packets transmitted, say *w*, called *fairness measurement window* in this work. As the computation of γ_i depends on *w*, the value of the Jain's index also depends on *w*, though *w* does not appear in the formula directly. Generally, the F_J value increases with *w*. *Absolute fairness* is achieved when F_J =1 while the *absolute unfairness* is achieved when F_J =1/*N*. As in [13], the index has been *averaged* over all *sliding* windows of *w* packets, which occur in the simulation run.

Figure 11 presents the Jain's index with reference to *w*. It is easy to see that the ECS greatly improves the fairness compared to IEEE 802.11. The reason that the large-EIFS problem leads to short-term unfairness is as follows. Consider that node B successfully transmits an upper-layer packet, and the chance that the node B gets control of the medium *again* is larger than that of node C, as node C suffers from the large-EIFS problem. On the other hand, once node C controls the medium, node C will also have high chance to transmit *consecutively*. Therefore, the large-EIFS problem results in short-term unfairness. However, since the large-EIFS problem occurs at nodes B and C with the same possibility, the long-term fairness between the two flows is ensured.

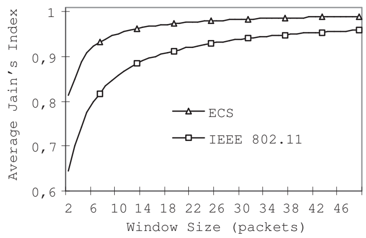

Fig. 11. Fairness Index Comparison for Scenario-3

Similarly, the short-term unfairness will also occur in the scenario of Figure 12 where the flow between nodes A and B is reversed compared to the scenario in Figure 10. However, in Figure 12, it is the receivers (i.e., nodes B and D) that suffer from the large-EIFS problem, rather the senders (i.e., nodes B and C) as in Figure 10.

Fig. 12. 4-nodes with Two Single-hop Flows

Scenario-4: Now we discuss the results for the scenario shown in Figure 13, which is similar to scenario-2 except that the distance between nodes B and C is increased to 400 meters. Therefore, nodes B and C are out of the TR but within the SR of each other, and they suffer from the small-EIFS problem as indicated by area 5 in the generalized scenario (Figure 6). Table IV presents the throughput results. The aggregate throughput under the IEEE 802.11 is 0.155 Mbps, which is much smaller than that under the ECS (i.e., 0.578 Mbps). However, even in under ECS, the aggregate throughput is still low compared to the maximum throughput (i.e., about 1.4 Mbps). The low throughput is not due to any deficiency in the ECS. To show this, we need to consider the scenario presented in Figure 14 where the SR is assumed to be equal to TR (i.e., 250 m) and the distance between nodes B and C is changed to 200 meters. Obviously, there is no imprecise-EIFS problem in the scenario of Figure 14. Moreover, if the ECS is able to solve the imprecise-EIFS problem in the scenario of Figure 13, the scenario should show the same performance as that of Figure 14, because when one flow is in progress, the other flow in both the scenarios will behave (e.g., defer) in the identical manner. This is verified by our simulation results.

Now we explain why the aggregate throughput is not close to 1.4 Mbps in the scenario of Figure 14 as well as in the scenario of Figure 13 (with ECS). Consider the situation that node A sends out a RTS to node B, and then node B sends back the CTS. During the transmission of the CTS, as node D is unaware of the CTS, it may send out a RTS, leading to a collision at node C. Therefore, node C cannot precisely determine that for how long it should defer its transmission. After the collision, node C will defer by an *standard* EIFS duration. Since the RTS/CTS handshaking is successful for the node A, it will transmit the Data frame. During the transmission, node D may initiate its RTS again. If the RTS arrives at node C after the EIFS deferment is over, node C will respond with CTS, which will collide with the Data frame from node A to B, and thus node B will discard the Data frame and defer by an EIFS. Now, for node D, since the RTS/CTS is successful, it will transmit the Data frame. The Data frame is likely to be destroyed by node B's CTS in a similar manner, resulting in substantial bandwidth wastage. Therefore, the throughput cannot be improved to 1.4 Mbps even when the imprecise-EIFS problem does not exist (as in the case SR=TR) or is completely solved (as in ECS). To improve the throughput further, one can incorporate the dual busy tone multiple access (DBTMA) protocol [9] into the ECS.

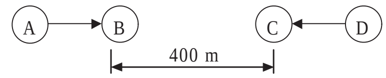

Fig. 13. 4-nodes with Two Single-hop Flows (Scenario-4)

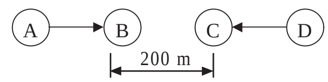

Fig. 14. Topology (SR=TR) Identical to Scenario-4 (ECS)

TABLE IV
THROUGHPUT COMPARISON FOR SCENARIO-4

Throughput (Mbps)	IEEE 802.11	ECS
A to B	0.079	0.290
D to C	0.076	0.288
Aggregate	0.155	0.578

Scenario-5: Now we discuss the scenario shown in Figure 15, which is similar to scenario-4 except that the direction of the flow between nodes C and D is reversed. Contrary to the situation that both nodes B and C suffer from the small-EIFS problem in the scenario-4, only one node (i.e., node C) suffers from the small-EIFS problem in this scenario. The average throughputs are presented in Table V. Under IEEE 802.11, the flow from A to B is completely starved while the other flow gets the entire bandwidth. Note that in our simulation the static routing is being used, and therefore this starvation is not caused by the failure of the routing discovery process as discussed in [14]. The reason for the starvation is as follows. As indicated in the generalized scenario (Figure 6), whenever a $FES(A, B)$ is in progress, the small-EIFS problem occurs at node C. In other words, whenever there is a Data transmission from A to B, it will always collide with node C's transmission, explaining why the flow from A to B is completely starved. Under the ECS, the flow from A to B gets about 0.075 Mbps, which is still much smaller than that of the other flow. However, again this is not due to the imprecise-EIFS problem, and therefore, the fairness cannot be ensured by the ECS. This can be validated by using an identical scenario (shown in Figure 16) where SR=TR, as done in the discussion of Scenario-4. In fact, the scenario of Figure 16 has been referred as the *asymmetrical information* scenario in [11]. As pointed out there, in such a scenario, the throughput of the flow from A to B can only get about 5% of the entire bandwidth, which is also true in our simulation. For the cause of the unfairness, please refer to [11]. To improve the fairness further, one can incorporate the Ordered Packet Scheduling

(OPS) algorithm [11] into the ECS.

Table V also shows that the aggregate throughput under the ECS improves marginally in comparison to the IEEE 802.11. This can be explained as follows. Under the IEEE 802.11, whenever node B sends back the CTS frame to node A, node C will defer by the EIFS value. However, the Data frame transmitted by node A will be discarded by node B as node C will send out a frame after it has deferred by the EIFS duration. Therefore, the bandwidth during the EIFS deferment at node C is wasted, explaining why the aggregate throughput under IEEE 802.11 is smaller than that in the ECS. Again, the improvement of ECS in the aggregate throughput should be much higher if a high-rate physical layer is used.

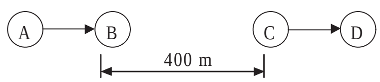

Fig. 15. 4-nodes with Two Single-hop Flows (Scenario-5)

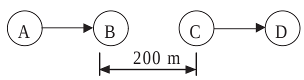

Fig. 16. Topology (SR=TR) Identical to Scenario-5 (ECS)

TABLE V

THROUGHPUT COMPARISON FOR SCENARIO-5

Throughput (Mbps)	IEEE 802.11	ECS
A to B	0.0	0.075
C to D	1.398	1.338
Aggregate	1.398	1.413

Scenario-6: Now we discuss the scenario shown in Figure 17. In this scenario, node C is out of the TR but within the SR of nodes A and B, while node D is out of the SR of nodes A and B. As indicated by the generalized scenario (Figure 6), the node C is in the area 6 and it suffers from the large-EIFS problem after the node B sends back an ACK frame to node A. The average throughputs for this scenario are presented in Table VI. It is easy to see that the throughputs under the IEEE 802.11 are very similar to those in scenario-1, as they exhibit the same problem. However, under the ECS, though the fairness substantially improves, the throughputs of the two flows are still not very close to each other (i.e., one is 0.672 Mbps while the other is 0.766 Mbps). Again, this unfairness is not due to any deficiency of ECS, which can be verified by using an identical scenario of Figure 18 with SR=TR.

To explain the unfairness even under ECS, let us consider the situation that the nodes A and C choose the same back-off timer before contending. Then, each of them initiates an RTS frame at the same time, resulting in a collision. Since node B is in the range (either SR or TR) of node C, it will detect this collision and thus it will not send CTS to node A. On the contrary, since node D is out of the range of node A, it will not detect the collision and therefore it will respond with a CTS to node C. In summary, whenever there is a collision between the RTSs of nodes A and C, only node C will be successful, explaining why the throughput of node C is greater than that of node A. Note that in the scenairio-1, whenever there is a collision between the RTSs of nodes A and B, neither receiver (i.e., B or C, respectively) will respond with a CTS, explaining why the ECS is very fair in that scenario.

By referring to the generalized scenario discussed in Section II.D, one can easily design some other scenarios (with two flows) to show the effects of the imprecise-EIFS problem and the advantage of our ECS. However, due to space limitation, we do not present any more such scenarios here.

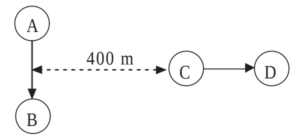

Fig. 17. 4-nodes with Two Single-hop Flows (Scenario-6)

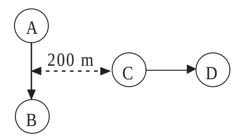

Fig. 18. Topology (SR=TR) Identical to Scenario-6 (ECS)

TABLE VI

THROUGHPUT COMPARISON FOR SCENARIO-6

Throughput (Mbps)	IEEE 802.11	ECS
A to B	1.161	0.672
C to D	0.254	0.766
Aggregate	1.415	1.438

B. Performance for Complex Scenarios

Now we study the scenarios that have more than two flows.

Scenario-7: Chain Topology

Figure 19 shows the chain topology with 10 nodes and 9 single-hop flows. The distance between any two neighboring nodes is 200 meters. The average throughputs are presented in Table VII. In the table, the flow ID is the same as the node ID of the flow's sender. We notice that the flow from node 5 to node 6 (i.e., flow with ID 5) is completely starved when IEEE 802.11 is used. However, under ECS, it gets about 0.008 Mbps, though it is still very small compared to the throughputs of other flows. To understand the performance more clearly, in Table VIII, we present the aggregate throughput, the standard deviation among the average throughputs of the flows, and the Jain's fairness index. Note that the Jain's Index is computed using the average throughputs of the flows, which means that the fairness measurement window is equal to the total number of packets transmitted in the entire simulation time.

From Table VIII, we find that, under ECS the standard deviation is smaller and the Jain's index is larger, showing that the ECS greatly improves the fairness compared to IEEE 802.11. However, we also notice that the improvement in fairness is at the cost of throughput, since the aggregate througput under ECS is smaller than that under IEEE 802.11. First of all, the chain topology can be viewed as a combination of topologies in figures 1, 12, and 15. As shown before, the imprecise-EIFS problem in these topologies will lead to unfairness only, but not the throughput degradation. Therefore, solving the imprecise-EIFS problem in the chain topology does not mean the aggregate throughput will improve, though the fairness should definitely improve, which is true in our results.

In fact, in the chain topology (Figure 19), the aggregate throughput is directly related to the contention in the network. To maximize the throughput, one can just starve all the flows except the flows 0, 4, and 8, which do not contend with each other, leading to an aggregate throughput of 4.2 Mbps (i.e., 1.4×3). It is easy to verify that, as long as the throughputs of the flows with IDs 1, 2, 3, 5, 6, and 7 are greater than zero, the aggregate throughput must be smaller than 4.2 Mbps, which is true under both IEEE 802.11 and ECS. Moreover, with the increase of the throughputs of these flows, the aggregate throughput will decrease. This explains why the aggregate throughput under ECS is smaller than that under the IEEE 802.11 as the flows 1, 2, 3, 5, 6, and 7 get more bandwidth under ECS. In fact, in multi-hop wireless ad hoc networks, maximizing throughput and achieving fairness are generally conflicting objectives as pointed out in [15].

We should also notice that, even the ECS does not distribute the bandwidth very fairly among the flows. Indeed, without deploying a scheduling algorithm (e.g., the one discussed in [15]) on the top of the MAC layer, it is very difficult for the MAC protocol itself to achieve the fairness among all the flows in a multi-hop scenario.

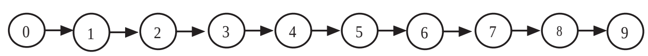

Fig. 19. Chain Topology: 10 Nodes with 9 Flows

TABLE VII

THROUGHPUT COMPARISON FOR THE CHAIN TOPOLOGY

Flow ID	0	1	2	3	4	5	6	7	8
IEEE 802.11	0.517	0.054	0.157	0.131	0.55	0.0	0.242	0.202	0.967
ECS	0.334	0.178	0.322	0.187	0.175	0.008	0.329	0.445	0.638

TABLE VIII

PERFORMANCE MEASURES COMPARISON (CHAIN TOPOLOGY)

Metrics	Aggregate	STD DEV	Jain's Index
IEEE 802.11	2.820	0.292	0.536
ECS	2.616	0.172	0.742

Scenario 8: Double-ring Topology

Figure 20 shows the double-ring topology with 16 nodes. The distance between the sender and the receiver of a flow is 200 meters. The diameter of the inner circle is also 200 meters, and thus the diameter of the outer circle is 600 meters. The angle between any two neighboring flows is 45 degrees.

The performance measures are presented in tables IX, and X. Again, the ID of a flow is the same as the node ID of the flow's sender. In contrast to the chain topology, the ECS greatly improves the aggregate throughput in this topology. As for the fairness, both the standard deviation and Jain's index under ECS are slightly greater than those under IEEE 802.11, which may convey very conflicting conclusions. If the standard deviation alone is considered, it seems that the ECS is more unfair than IEEE 802.11. On the other hand, the Jain's index is indicating that the ECS is more fair. Because the throughputs under the two schemes differ by an order of magnitude, a larger standard deviation does not imply that ECS is more unfair. Therefore, standard deviation is not always a good measure of fairness, while the Jain's index is.

We have also conducted the simulation for some more complex scenarios (e.g., grid topologies). It is found that the ECS always improves the performance of IEEE 802.11.

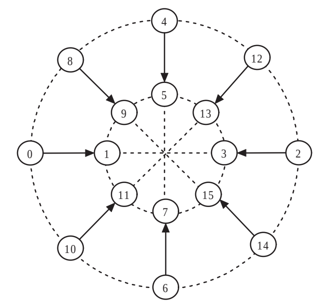

Fig. 20. Double-ring Topology: 16 Nodes with 8 Flows

TABLE IX

THROUGHPUT COMPARISON FOR THE DOUBLE-RING TOPOLOGY

Flow ID	0	2	4	6	8	10	12	14
IEEE 802.11	0.0169	0.0214	0.0197	0.0195	0.0207	0.0206	0.0197	0.0209
ECS	0.1586	0.1587	0.1542	0.1517	0.1641	0.1627	0.1663	0.1483

TABLE X

PERFORMANCE MEASURES COMPARISON (DOUBLE-RING TOPOLOGY)

Metrics	Aggregate	STD DEV	Jain's Index
IEEE 802.11	0.1594	0.0013	0.9957
ECS	1.2646	0.0059	0.9986

Performance under Imperfect ECS

In the previous subsections, we have demonstrated that ECS greatly improves both the throughput and fairness performance under the assumption that the ECS can always correctly detect the *type* of an erroneous frame. However, as discussed in Section III, there are certain situations where the ECS may not perfectly detect the type of an erroneous frame. We study the impact of this under a simple probabilistic model. In particular, we assume that ECS cannot detect the type of a frame with a probability ρ. As stated before, the standard EIFS value is used in this case.

We first show the performance for Scenario-1 to study the impact on fairness. Figure 21 shows that when the probability ρ increases, the fairness between the two flows becomes worse. The fairness performance becomes very similar to IEEE 802.11 when ρ becomes one.

We now study the impact on the aggregate throughput in Scenario-2. Figure 22 shows that when the probability ρ increases, the aggregate throughput becomes smaller, and the aggregate throughput becomes the same as IEEE 802.11 when ρ becomes one.

The above two scenarios show that the performance of ECS degrades linearly as the probability ρ approaches unity. In practice, we expect the ρ to be small where the benefits of ECS are retained to a large extent.

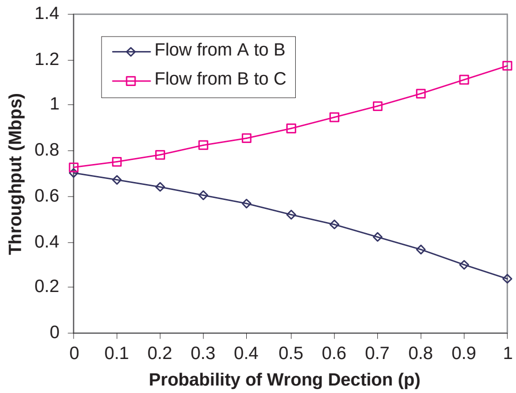

Fig. 21. Fairness under Imperfect ECS (Scenario-1)

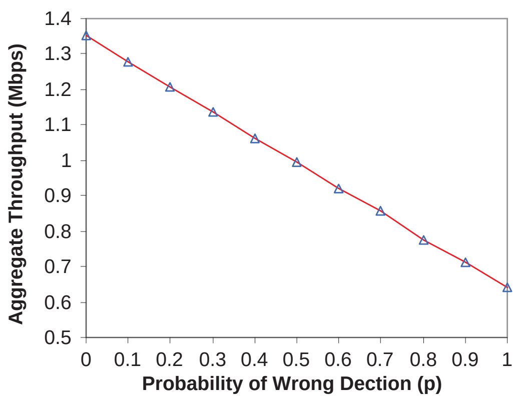

Fig. 22. Throughput under Imperfect ECS (Scenario-2)

V. Conclusions

In this paper, we have addressed the following issue: *whenever a node detects an erroneous frame (e.g., a sensing range frame or a transmission range frame with a wireless error), how long should the node defer its transmission to prevent the potential interference on the ongoing transmission over the shared medium*? This issue has not been addressed in the literature. In IEEE 802.11, whenever a node detects an erroneous frame, it always defers by a fixed duration (represented by EIFS). Due to this fixed EIFS value, we showed that two problems arise: small-EIFS problem and large-EIFS problem, which lead to considerable unfairness and throughput degradation. In order to solve the problems, we have proposed an enhanced carrier sensing (ECS) mechanism. In the ECS, the lengths of the frames are made different. Based on the length of a frame observed on the medium, the *type* of the frame can be detected, and the node defers the transmission for a duration accordingly, rather than by the fixed duration. In order to obtain the length of an erroneous frame, the ECS utilizes the information provided by the physical layer. However, the ECS does not need any enhancement at the physical layer or on the cooperation between the MAC and physical layers. Therefore, the ECS can be easily incorporated into IEEE 802.11. Our extensive simulation results have shown that the ECS improves the fairness as well as throughput drastically.

Compared to the large number of published works that focus on the collision avoidance (CA) or contention resolution (CR) algorithm to improve the fairness or throughput in the CSMA/CA-based MAC protocols (e.g., IEEE 802.11), our work is unique as we have focused on the carrier sensing (CS) mechanism to improve both the fairness and the throughput, which is not available in the literature.

Further improvement in the performance is possible by refining the contention resolution and collision avoidance techniques and integrating them with our enhanced carrier sensing (ECS) scheme.

Appendix
Average Throughput Computation in Scenario-1

In the scenario exhibited in Figure 1, using the simulation, the flow from node A to B gets about 0.25 Mbps while the flow from B to C gets about 1.15 Mbps, therefore the proportions are as follows:

$$Proportion(flow\ from\ A\ to\ B) = 0.25/1.4 \approx 0.18$$
$$Proportion(flow\ from\ B\ to\ C) = 1.15/1.4 \approx 0.82$$

In this appendix, using an analytical model we explain why there is such a large difference in the throughputs of the two flows. Since nodes A and B know the state of the medium before initiating a RTS, the collision probability is very small and the contention window (CW) is very unlikely to be greater than CW_{min} (i.e., 31). Neglecting the collision probability, the system can be modeled as a two-states Markov chain, which is illustrated in Figure 23.

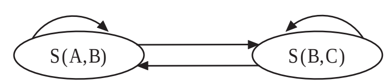

Fig. 23. Two-states Markov Chain for Scenario-1

The state $S(A,B)$ represents that the $FES(A,B)$ is in progress while the state $S(B,C)$ corresponds to the $FES(B,C)$. To find the state probabilities of $S(A,B)$ and $S(B,C)$, we only need to know the transition probabilities, which must satisfy the following conditions:

$$Pr(S(B,C)|S(A,B)) = 1 - Pr(S(A,B)|S(A,B))$$
$$Pr(S(B,C)|S(B,C)) = 1 - Pr(S(A,B)|S(B,C))$$

Therefore, we need only compute two transition probabilities, i.e., $Pr(S(A,B)|S(A,B))$ and $Pr(S(A,B)|S(B,C))$.

Let us first compute $Pr(S(A,B)|S(A,B))$. From Figure 3(a) we notice that after the completion of $FES(A,B)$, nodes A and B will first defer their transmission by DIFS. Then, node A generates a *new* random back-off timer while node B *does not*, since it already has a non-zero back-off timer that was generated in the previous competition round. We call this nonzero back-off timer as the *remaining* back-off timer. If the newly generated back-off timers at node A is detonated by I while the *remaining* back-off timer at node B is J, then the condition under which node A will get access to the medium *again* after the completion of the $FES(A, B)$ is that, the I is smaller than the J. Therefore,

$$Pr(S(A, B)|S(A, B)) = Pr(I < J) \tag{1}$$

where I observes the uniform distribution over the range [0, 31], but it is not trivial to get the distribution of J. If we denote the random back-off timers generated in the *previous* round at nodes A and B as X and Y respectively, then J is equal to $(Y - X)$ and the distribution can be derived as follows:

$$Pr(J = j) = \frac{2 \times (32 - j)}{32 \times 31} \quad j \in [1, 31] \tag{2}$$

Using the conditional probability, the equation (1) becomes,

$$\begin{aligned} &Pr(S(A, B)|S(A, B)) \\ &= \textstyle\sum_{j=1}^{31} Pr((0 \le I < j)|(J = j)) \times Pr(J = j) \end{aligned}$$

Since I and J are *independent* to each other, the above formula becomes,

$$Pr(S(A, B)|S(A, B)) = \textstyle\sum_{j=1}^{31} Pr(0 \le I < j) \times Pr(J = j)$$

Therefore, the $Pr(S(A, B)|S(A, B))$ is computed as follows:

$$Pr(S(A, B)|S(A, B)) = \textstyle\sum_{j=1}^{31} j/32 \times Pr(J = j) = 0.3438$$

Let us now calculate the $Pr(S(A, B)|S(B, C))$. After the completion of $FES(B, C)$, node B first defers by DIFS and then generates a *new* random back-off timer. On the other hand, since node A suffers from the *large-EIFS* problem (see Figure 3(b)), it will first defer by EIFS rather than by DIFS and then resume to count down the frozen back-off timer. The only condition under which node A will get access to the medium after the completion of $FES(B, C)$ is that, the J is larger than $(EIFS - DIFS)$ and the I is smaller than $(J - (EIFS - DIFS))$. Since the back-off timer is *integral* multiple of slot-time (i.e., 20 us in the DSSS [10]), the $(EIFS - DIFS)$, which is equal to 314 μs, should be rounded to 16 slots. Therefore,

$$\begin{aligned} &Pr(S(A, B)|S(B, C)) \\ &= Pr(\ (J > 16)\ and\ (I < J - 16)\ ) \end{aligned}$$

Using conditional probability, it becomes,

$$\begin{aligned} &Pr(S(A, B)|S(B, C)) \\ &= \textstyle\sum_{j=17}^{31} Pr((0 \le I < j - 16)|(J = j)) \times Pr(J = j) \end{aligned}$$

Contrary to the situation of computing $Pr(S(A, B)|S(A, B))$, in this case J observes the uniform distribution while I observes the distribution indicated by equation (2). Therefore, the $Pr(S(A, B)|S(B, C))$ is computed as follows:

$$\begin{aligned} &Pr(S(A, B)|S(B, C)) \\ &= \textstyle\sum_{j=17}^{31} Pr(0 \le I < j - 16) \times Pr(J = j) = 0.1764 \end{aligned}$$

Therefore, after that a $FES(B, C)$ is completed, the probability that node A can get control of the medium is very small (i.e., 0.1764), showing the unfairness caused by the large-EIFS problem at node A.

Once we have obtained all the transition probabilities, it is easy to get the state probabilities as follows:

$$\begin{aligned} Pr(S(A, B)) = 0.21 \\ Pr(S(B, C)) = 0.79 \end{aligned}$$

We notice that the analytical results are quite *close* to the simulation results, and thus verify our simulation.

References

[1] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LANs," ACM SIGCOMM, 1994.

[2] V. Bharghavan, "Performance Evaluation of Algorithms for Wireless Medium Access," IEEE performance symposium, 1998.

[3] F. Cali, M. Conti, E. Gregori, "Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit," IEEE Journal on Selected Areas in Communications, December 2000, pp.785-799.

[4] —- "IEEE 802.11 Protocol: Design and Performance Evaluation of an Adaptive Backoff Mechanism," IEEE Transactions on Networks, September 2000, pp.1774-1786.

[5] D. Chiu and R. Jain, "Analysis of the Increase/Decrease Algorithms for Congestion Avoidance in Computer Networks," Journal of Computer Networks and ISDN, Vol. 17, No. 1, June 1989, pp. 1-14.

[6] CMU Monarch Group. CMU Monarch Extensions to NS, http://www.monarch.cs.cmu.edu/.

[7] Z. Fang, B. Bensaou, Y. Wang, "Performance Evaluation of a Fair Back-off Algorithm for IEEE 802.11 DFWMAC," ACM MOBIHOC, 2002.

[8] C. L. Fullmer and J. J. Garcia-Luna-Aceves, "Floor Acquisition Multiple Access for Packet-Radio Networks," ACM SIGCOMM, 1995.

[9] Z.J. Haas, J. Deng. "Dual Busy Tone Multiple Access (DBTMA)-A Multiple Access Control Scheme for Ad Hoc Networks," IEEE Transaction on Communications, June 2002, pp.975-985.

[10] IEEE, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," IEEE 802.11 standards, June 1999.

[11] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, E, Knightly, "Ordered Packet Scheduling in Wireless Ad Hoc Networks: Mechanisms and Performance Analysis," in ACM MOBIHOC, 2002.

[12] P. Karn, "MACA - a new channel access method for packet radio," ARRL/CRRL Amateur Radio 9th Computer Networking Conference, 1990.

[13] C.E. Koksal, H. Kassab, H. Balakrishnan, "An Analysis of Short-Term Fairness in Wireless Media Access Protocols," ACM SIGMETRICS, 2000.

[14] Z.F. Li, A. Gupta, S. Nandi, "Study of IEEE 802.11 Fairness and its Interaction with Routing Mechanism," IFIP MWCN, 2003.

[15] H. Lou, S. Lu, V. Bharghavan, "A New Model for Packet Scheduling in Multi-hop Wireless Networks," ACM MOBICOM, 2000.

[16] T. Nandagopal, T. Kim, X. Gao, V. Bharghavan, "Achieving MAC Layer Fairness in Wireless Packet Networks," ACM MOBICOM, 2000.

[17] N.H. Vaidya, P. Bahl, S. Gupta, "Distributed fair scheduling in a wireless LAN," ACM MOBICOM, 2000.

[18] K.X. Xu, M. Gerla, S. Bae, "How Effective is the IEEE 802.11 RTS/CTS Handshke in Ad Hoc Networks?", IEEE Globecom, 2002.