Achieving MAC Fairness in Wireless Ad-hoc Networks by Cooperation between Sender and Receiver

Zhifei Li
School of Computer Engineering
Nanyang Technological University
Singapore, 639798

Anil K. Gupta
School of Computer Engineering
Nanyang Technological University
Singapore, 639798

Sukumar Nandi
Dept. of Computer Sc. & Engineering
Indian Institute of Technology, Guwahati
India, 781039

***Abstract*— Achieving MAC layer fairness in wireless ad hoc networks is a very challenging issue. In this paper, we first show that IEEE 802.11 exhibits substantial unfairness due to the concealed information problem (explained in this paper later), imprecise collision detection mechanism, and high contention. To achieve MAC fairness, we introduce a Fair MAC protocol using Cooperation between Sender and Receiver (FMAC/CSR). In FMAC/CSR, each node, in a distributed manner, estimates the *number of active nodes* within the contention range, as well as its *actual bandwidth share*. The estimated number of active nodes is used by a node to dynamically determine its fair share. In the IEEE 802.11, due to unfairness, the actual share of a node may deviate from its fair share. To achieve fairness, based on the deviation between fair share and actual share, a node enters one of the three modes, *aggressive*, *restrictive*, and *normal*. The sender controls the access to the medium in a differentiated manner according to the mode it enters. However, due to the concealed information problem, the estimation at the sender may not be precise and thus the sender cannot contend in a fair manner. Therefore, we propose the receiver-coordination mechanisms including over-use notification and Request RTS (RRTS), which exploit the information available only at the receiver to help the sender access the medium more fairly. Extensive simulation results show that the proposed FMAC/CSR scheme substantially improves the MAC fairness without unduly degrading the throughput.**

I. INTRODUCTION

Recently, wireless ad hoc networks have attracted considerable research interest. As IEEE 802.11 [10] is the de facto standard for Wireless LANs, most of the research work on wireless ad hoc network adopt it as the MAC layer. IEEE 802.11 defines two MAC protocols: Point Coordination Function (PCF) and Distributed Coordination Function (DCF). However, only the DCF is used in the ad hoc networks. DCF is a CSMA/CA-based MAC protocol in which the sender plays a major role in the contention of the medium. In wireless ad hoc networks, since a sender may not have precise information of the contention on the medium, fairness in accessing the medium is one of the most challenging issues.

Based on the *length* of the time over which we observe the system, the fairness can be defined on a short-term basis and on a long-term basis. The short-term *fairness* automatically gives rise to the long-term fairness, but not the vice versa [13]. On the other hand, the long-term *unfairness* implies the short-term unfairness, but not the vice versa. Both long-term and short-term unfairness have great impact on the system performance (e.g., Quality of Service).

Fairness can be achieved through two different ways. One is to design a scheduler (e.g., [14]), which is overlaid on the top of the MAC layer, to guaranty a node's fair share proportional to its weight. The other method is to achieve the fairness at the MAC layer itself (e.g., [16]). We here follow the second method. We first show that the IEEE 802.11 exhibits substantial short-term unfairness due to the concealed information problem (explained in this paper later), imprecise collision detection mechanism, and high contention. To achieve fairness, we then propose a general MAC fairness framework, which includes three components: fairness model to define fair shares, compensation model to compensate for the over-use and under-use with respect to the fair shares, and distributed algorithm to realize the fairness. Our proposed distributed algorithm, called FMAC/CSR, exploits information available at the sender and receiver to control access to the medium. In the FMAC/CSR, every node, in a *distributed* manner, estimates the number of *active* stations (say *n*) within the contention range, as well as the actual bandwidth share (say w_x) received by the node. The estimate of *n* is a good indication of the contention degree, and therefore, it can be used by the node to dynamically determine its fair share (say ϕ_x). Due to the unfairness in IEEE 802.11, the actual share (i.e., w_x) may deviate from the fair share (i.e., ϕ_x). To make the w_x as close to ϕ_x as possible, when contending for the medium, the node enters one of the three modes, *aggressive*, *restrictive*, and *normal*, indicating how the node should behave when contending for the medium. According to the mode a sender enters into and the amount by which w_x differs from ϕ_x, the sender accesses the medium with different priority. However, due to the concealed information problem, the estimation at a sender may not reflect the actual contention conditions of the medium and thus the sender cannot contend in a fair manner. Therefore, we propose receiver-coordination

mechanisms including over-use notification and Request RTS (RRTS), which exploit the information available only at the receivers to help the senders make more rational decisions. Extensive simulation results show that the proposed FMAC/CSR substantially improves the fairness without unduly degrading the throughput.

The remainder of the paper is organized as follows. In Section II, we describe the basic technique of IEEE 802.11 and illustrate the unfairness. We then, in Section III, present a general MAC fairness framework. The estimation algorithm and the FMAC/CSR are presented in Sections IV and V, respectively. We then present the simulation results in Section VI. The related work is reviewed in Section VII, and the paper is concluded in Section VIII.

II. Unfairness in IEEE 802.11

A. Basic Techniques in IEEE 802.11 DCF

The DCF defines two methods for accessing the medium: the two-way handshake and the four-way handshake. In the two-way handshake, the sender first transmits a Data frame to the receiver, which responds with an ACK frame if it receives the Data frame correctly. On the other hand, in the four-way handshake, the sender first sends out a Request To Send (RTS) frame. In response to this request, the receiver sends back a Clear To Send (CTS) frame if the medium is determined to be idle. Then, the sender sends out the Data frame and the receiver responds with an ACK frame. To cope with the common hidden-terminal problem in ad-hoc wireless networks, we assume that the four-way handshaking is used.

The IEEE 802.11 adopts the well-known Binary Exponential Back-off (BEB) algorithm as its Contention Resolution (CR) mechanism, which is described as follows. Every node maintains a Contention Window (CW) and a back-off timer. Before every transmission, the node first defers by a back-off timer, which is generated according to equation (1), unless the back-off timer already contains a non-zero value, in which case it is unnecessary to generate a new random back-off timer.

$$BackoffTime = Random() \times SlotTime \quad (1)$$

The *SlotTime* is specified by the physical layer, and the *random* value is uniformly distributed over the range [0, CW]. For the first transmission attempt of a packet, the CW will be set to CW_{min}. Whenever a retransmission is initiated, the CW is doubled. When a retry limit is reached, the CW will be reset to CW_{min}. The CW is also reset to CW_{min} whenever a transmission is successful.

When a node (say *H*) is transmitting a packet, the other nodes *freeze* their back-off timers. After node *H* completes transmission of the packet and thus the medium becomes idle, all the contending nodes first defer for a DCF Inter-Frame Space (DIFS) period. Then, node *H* generates a new random value from its CW and backs off before it initiates another transmission. On the contrary, the other nodes simply resume to count down from their *frozen* back-off timers. Due to the *freezing mechanism*, node *H* may transmit several packets *consecutively* before another node's back-off timer is reduced to zero, leading to short-term unfairness. Contrary to a successful transmission, when a collision occurs, all the colliding nodes will generate a new random value.

B. Unfairness Due to Concealed Information Problem

Since the CSMA/CA-based MAC protocols (e.g., IEEE 802.11) normally adopt two-direction handshakes, when a flow contends for the medium, the contention is at both the sender and the receiver. Therefore, two flows are contending with each other if either the sender or the receiver of one flow is within the transmission range[1] of the sender or the receiver of the other flow [15].

We now try to identify the possible scenarios, which have two contending flows and exhibit short-term or long-term unfairness. For convenience, we define $F(X, Y)$ as a single-hop flow between nodes X and Y, where node X is the sender and node Y is the receiver. Let us consider the situation that two flows, say $F(S_A, R_A)$ and $F(S_B, R_B)$, contend for a common medium. We assume that S_A, R_A, S_B, R_B are different nodes. Figure 1 illustrates the relationship between node S_A and *flow* B. In the figure, whenever two nodes are in the range of each other, there is a line connecting them. It is easy to see that there are four possibilities as far as the relationship between node S_A and flow B is concerned. This is also true for the relationship between node R_A and flow B. As a result, the number of possible scenarios having different relationships between *flows* A and B is sixteen (4×4). However, in the scenario that both S_A and R_A are in the range of neither sender nor receiver of flow B, flows A and B are not contending any more. Therefore, the number of possible scenarios reduces to fifteen. Moreover, as shown in [11] as well as in our simulation, in the scenarios that the senders are in the range of each other (the cases labelled as (2) and (4) in Figure 1), the flows can contend very fairly (except the two scenarios discussed in sections II-C and II-D). There are eight (2×4) scenarios where the two senders are within range of each other. As a result, seven $(15-8)$ scenarios remains to be studied. Based on the relationship between the sender of one flow and the receiver of the other flow, the seven scenarios can be further classified into three categories. In the *first* category, none of the senders is within the range of the receiver of the other flow. The scenario-1 shown in Figure 2 belongs to this category. The *second* category is that only one of the senders is within the range of the receiver of the other flow. The scenario-2 and -3 shown in Figure 3 where S_B is within the range of the R_A belong to such a category. Note that two other scenarios, where S_A is within the range of the R_B, also belong this category. However, we do not show them in the figure as they will have the same problem as that in scenario-2 and -3. In the *third* category, both the senders are within the range of the receiver of the other flow. The scenario-4 and -5 in Figure 4 belong to this category.

However, if the two flows just involve *three* nodes, the only scenario where the senders are not within range of each

[1]It should be the sensing range if the sensing range is greater than the transmission range.

other is scenario-6 shown in Figure 4, which is also known as the hidden-terminal scenario. Obviously, if the two flows just involve *two* nodes, the two senders are always within the range of each other. In these scenarios, though the unfairness may still occur, we do not consider them since the fairness can simply be achieved by using fair queuing algorithms developed for the wire-line networks.

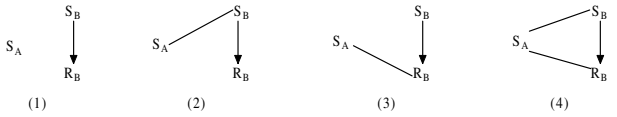

Fig. 1. Relationship between node S_A and Flow B

Now, we will show how the unfairness takes place in the above scenarios.

Short-term Unfairness in Scenario-1: In Figure 2, the distance between the senders of the two flows is three hops, which is in fact the *maximum* distance between two senders if they are contending for the same medium. Now we explain the reasons of the unfairness in this scenario (though the simulation results are presented in Section VI-C). Consider the situation that flow A is in progress and thus node R_B is not able to respond to any request from node S_B. However, since node S_B does not know about the ongoing transmission of flow A, S_B may futilely retry, resulting in a large CW at the node. As for the node S_A, after it transmits the packet, it will reset its CW and contend for the medium again. Since the CW at node S_B becomes very large, node S_A may transmit several packets *consecutively* before node S_B gets control of the medium, leading to short-term unfairness. However, several mechanisms incorporated in the IEEE 802.11 prevent flow B from starving *completely*, such as: *(i)* after every packet transmission, node S_A will back-off before initiating another transmission, which gives node S_B a chance to contend for the medium; *(ii)* the CW at node S_B will be reset to CW_{min} after the retry limit is reached. Moreover, once node S_B gets control of the medium, it will capture the medium for a long time in a similar manner. As a result, the long-term fairness between these two flows are ensured in IEEE 802.11[2].

Fig. 2. Category 1: Scenario-1

Long-term Unfairness in Asymmetrical Information Scenarios: In this category, the sender, who is in the range of the receiver of the other flow, can get more information of the medium than the other sender does. This has been referred as the asymmetrical information problem [11], which results in substantial long-term unfairness as explained below. Whenever S_A has complted transmission of a packet, both S_A and S_B begin to contend for the medium. In such a situation, the only condition that S_A can win is that R_A responds with a CTS before S_B begins to send out an RTS. Otherwise, S_B will definitely win the contention even it begins to send out the RTS later than S_A does. This is obviously unfair for flow A.

[2]According to the results presented in [2], in such a scenario, flow B will be completely starved in the MACAW protocol.

Now let us consider the situation that flow B is transmitting a packet. Node S_A cannot even identify when the transmission of the packet will be end. Therefore, during the period that S_B is transmitting, S_A will contend futilely, resulting in a large CW, leading to unfairness for S_A. In summary, after every transmission (transmitted either by flow A or B), S_A will always be treated unfairly, resulting in long-term unfairness. Note that though the receivers in the scenario-2 and -3 have different information about the contention, the two scenarios will show the same performance since the receiver in IEEE 802.11 does not play an active role in the contention.

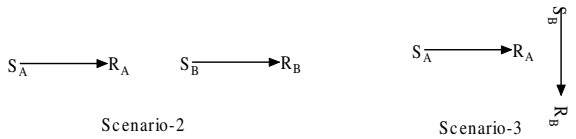

Fig. 3. Category 2: Asymmetrical Information Scenarios

Short-term Unfairness in Hidden-terminal Scenarios: In the hidden-terminal scenarios (Figure 4), if the four-way handshake is used, and once the RTS/CTS has been completed successfully, the hidden-terminal problem does not arise any more. For example, once node R_A sends back a CTS to node S_A, node S_B overhears this CTS and defers its transmission, avoiding collision. However, the four-way handshake cannot *eliminate* the hidden-terminal problem, as the RTSs sent by the two hidden senders may still collide, unless the following condition is satisfied,

$$| Z | > Len = TxTime(RTS) + SIFS \qquad (2)$$

where *Z* is the difference between the back-off timers at the two hidden senders, and *Len* is equal to the transmission time of RTS plus a Short Inter-Frame Space (SIFS), which is about 19 slots when the DSSS [10] is used. It is easy to see that the condition is difficult to satisfy when the CWs are small (e.g., 31).

Now let us explain how the hidden-terminal problem causes short-term unfairness (the simulation results are presented in Section VI-A). Consider the situation that the CWs at both the senders are very small (e.g., 31). As discussed above, under such situation the condition illustrated in equation (2) is difficult to satisfy, and thus the RTSs of the senders may collide. The collision may occur several times until the CWs are large enough to allow either sender to get control of the medium. In particular, one of the two sender (say, node S_A) may select a small back-off time from its CW, while the other sender (i.e., S_B) selects a large value. The difference between the two values may be large enough to satisfy the condition, and thus node S_A can successfully transmit a packet. Once the packet transmission is completed, node S_A resets its CW and backs-off before initiating another transmission. However, the *remaining* back-off timer at node S_B may be large compared to the back-off timer at node S_A, which is drawn from the range [0, CW_{min}]. In that case, node S_A may transmit several more packets before node S_B's back-off timer decrements to a small value.

Whenever the back-off timer at node S_B becomes small, node S_B contends for the medium. However, as the CW at S_A is equal to CW_{min}, the contention is most likely to result in a collision. After the collision, node S_A doubles its CW from CW_{min} whereas S_B doubles its CW from a larger value (at least 63). Therefore, node S_A is more likely to get control of the medium *again*. This is obviously unfair for S_B. Moreover, this process may repeat several times, leading to starvation at node S_B for a considerable period (compared to the time needed for a packet transmission). However, the mechanisms of IEEE 802.11 discussed before prevent flow B from starving *completely*, and thus ensure long-term fairness between the two flows. Again, the three scenarios of Figure 4 show the same performance though the receivers have different degrees of information of the medium.

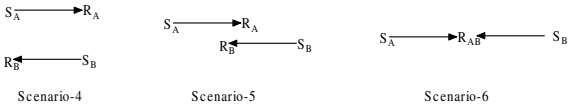

Fig. 4. Category 3: Hidden-terminal Scenarios

Concealed Information Problem: From all the above scenarios, we notice that the main problem leading to unfairness in the CSMA/CA protocols is that the senders cannot obtain precise information about the contention for the medium. In other words, the state information is *concealed* from the sender, and therefore we call it the *concealed information problem*. Here, the information a sender needs includes: *(i)* when does the contention period (i.e., the duration after the ACK frame and before the next RTS frame on the medium) occur, and *(ii)* if the contention period is identified, whether some other sender has already sent out a frame during the period.

In the scenario-1 (Figure 2), the senders are lacking in both kinds of information. In the asymmetrical information scenarios (Figure 3), one sender suffers from the concealed information problem, while the other sender does not, resulting in *long-term unfairness* between the two flows. In the hidden-terminal topologies (Figure 4), though the senders can identify the contention period, the transmission of the RTS by one sender (e.g, node S_A) is *concealed* from the other sender (i.e., S_B).

C. Long-term Unfairness Due to Imprecise Collision Detection Mechanism

In this subsection, we will show that the collision detection mechanism used in IEEE 802.11 results in long-term unfairness even when the senders are in the range of each other. Figure 5 presents such a scenario. Let us consider the situation that the nodes S_A and S_B choose the same back-off value before contending. Then, each of them initiates an RTS at the same time, resulting in a collision. Since node R_A is in the range of node S_B, R_A will detect this collision and thus it will not send CTS to node S_A. On the contrary, since node R_B is out of the range of node S_A, R_B will *not* detect the collision and therefore it will respond with a CTS to node S_B. In summary, whenever there is a collision between the two RTSs, only node S_B will be successful, explaining unfairness for flow A.

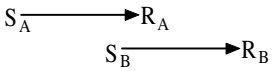

Fig. 5. Scenario-7: Imprecise Collision Detection Mechanism

D. Short-term Unfairness Due to High Contention

Now we will show that, in the scenarios with high contention, the short-term unfairness may occur even when all the senders are within range of each other and all the nodes have the same understanding when a collision occurs. For example, in the scenario shown in Figure 6, there are five flows and all the six stations are in the range of each other. In order to explain why there is short-term unfairness (the simulation results presented in Section VI-E), we need to look deeper in the BEB that is being employed to resolve collisions. The probability of collision depends upon the number of active nodes (say *n*) within the contention range as well as the CWs at the nodes. A node is referred to be *active* whenever it has at least one packet waiting to send. As *n* increases, the collision probability also increases. On the other hand, as the CW increases, the collision probability decreases. The BEB uses collisions to gauge the contention degree and dynamically adjusts the CWs to resolve the collisions. In particular, when the contention is very high while the CWs are very small, collision(s) occur and the colliding nodes increase their CWs. Hopefully, some of the colliding nodes generate large back-off timers and thus they defer the contention for the medium. This indirectly reduces the contention degree and thus resolves the collision effectively.

The deferring nodes will join in the contention in the future, and this time, some *other* nodes may defer their transmission (due to collisions), and that is why the *long-term* fairness is ensured. However, during a short-term, due to *randomness*, the same nodes may collide consecutively while other nodes do not experience any collisions, leading to short-term unfairness. This is particularly true when the contention is very high as collisions are more likely to happen. In summary, though the BEB is effective in resolving collisions, it results in shot-term unfairness when the contention is high.

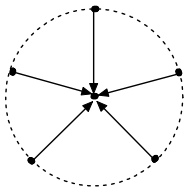

Fig. 6. Scenario-8: High Contention

So far, we have systematically identified the possible scenarios that show unfairness due to concealed information (Figures 2, 3, and 4), imprecise collision detection mechanism (Figure 5), or high-contention (Figure 6).

E. Impact of Unfairness

While the impact of long-term unfairness is obvious, the short-term fairness is also very important for adaptive traffic

(e.g., TCP kind of traffic) and the delay- or jitter-sensitive traffic as discuss in [13] and [20]. Moreover, the short-term unfairness may greatly affect the behavior of on-demand routing protocols. For example, if a node cannot transmit a packet after several retries due to the short-term unfairness at the MAC layer (e.g., in scenarios of Figures 2, 3, and 4), the MAC layer will discard the packet and convey this event to the routing protocol. As the routing protocols (e.g., AODV) interpret the discarding of a packet as an indication of link-breakage, and thus the routing protocol will discard all the packets in the queue and initiate a new route discovery process, though the link is still available. Even worse, in such a situation, the route discovery process is difficult to be successful due to the unreliable nature of broadcast packets in IEEE 802.11 and due to the exponential back-off of the routing discovery itself. Therefore, a flow may be starved for a considerable time. It is clear that it is extremely important to achieve both short-term and long-term fairness at the MAC layer.

III. General Framework to Achieve MAC Fairness

In order to achieve MAC fairness in wireless ad-hoc networks, we propose that the framework should include the following three components. *Firstly*, we should define a fairness model, which determines how much share a user should get to maintain fairness, and how long the duration be over which the fairness is measured. *Secondly*, since the actual share of an active node may deviate from its fair share, a compensation model is needed to compensate the share of the node based on its past usage of the medium. *Thirdly*, we should have distributed algorithm(s) in place, which tries to make the actual share of a node close to its fair share in every fairness measurement duration.

A. Fairness Model

The objective of a fairness algorithm is to ensure that a node gets a fair share (say, ϕ_x for node x) over every certain duration (say T_{cyc}). The value of T_{cyc} determines the duration over which the fairness is desired. For example, if T_{cyc} is large, the algorithm is aiming to achieve the long-term fairness, which however does not necessarily imply short-term fairness. On the other hand, if T_{cyc} is small, the objective is to achieve the short-term fairness, which automatically gives rise to the long-term fairness. If the fairness is defined as the equal share among *n* active nodes (obviously *n* varies with time), it is reasonable to assign T_{cyc} with the time required to transmit *n* packets. Moreover, if all the packets are assumed to have the same length, the T_{cyc} can simply be replaced by a *transmission window* (say W_{cyc}), which is equal to *n*. Obviously, T_{cyc} or W_{cyc} is more complex to define if the fairness model needs to provide differentiated service among the active nodes. Moreover, how to decide the fair share (ϕ_x for node x) is in itself an issue. At a *higher* layer, the value of ϕ_x should be determined based on the application requirement. However, as our focus is to achieve fairness in the contention for the shared medium, we simply assume that all the applications have the same requirements, and hence the weights are equal. Therefore, at the MAC layer, it is reasonable to assign ϕ_x with $1/n$ when *n* nodes are active, implying that every *active* node should transmit *exactly* one packet whenever *n* packets are transmitted over the medium. In summary, this can be formalized as,

$$\begin{cases} T_{cyc} = W_{cyc} = n \\ \phi_x = 1/n \end{cases} \quad (3)$$

T_{cyc} and ϕ_x, defined as above, change with the contention degree (i.e., *n*), rather than being pre-defined as in most of the literature (e.g., [6]), and thus are very adaptive to the dynamic network conditions.

B. Compensation Model

If the actual share (say, w_x for node x) is always equal to ϕ_x during every window W_{cyc}, a system using the above fairness model behaves like a dynamic TDMA protocol, which has the *ideal* fairness. However, the w_x may deviate from ϕ_x, especially in the wireless ad-hoc networks due to the short-term or long-term unfairness discussed in Section II. In particular, during a given window of length W_{cyc}, a node may transmit more than one packet. We refer to this as the *over-use* of the medium by the node. On the other hand, if a node does not transmit any packet during the window, we call it *under-use*. Naturally, *normal-use* refers to the case when a node transmits *exactly* one packet in the window. To compensate for the over-use and under-use in the previous window, and thus make w_x as close to ϕ_x as possible, an active node should adjust its rate as early as possible. Specifically, from the contention viewpoint, an active node should be in one of the three modes: *aggressive*, *restrictive*, and *normal* at any given time. If a node has *under-used* during the previous window, it should give itself more opportunity in contending for the medium, and thus should enter the aggressive mode. On the other hand, if a node has *over-used* the medium, it should be in the restrictive mode. However, if a node gets its *fair* share, it should operate in the normal mode.

It is clear that the above approach is based on the knowledge of *n* (the number of active nodes) and w_x (the actual share of node x). However, in the ad hoc networks, every node has to dynamically estimate these two values in a *distributed* manner. Moreover, due to distributed nature of the wireless ad-hoc networks, how to design a medium contention algorithm that can realize the compensation is a demanding challenge. The above two issues are discussed in sections IV and V, respectively.

IV. Estimation Algorithm

Whenever a node transmits a *packet* over the medium using the four-way handshake, other nodes within the contention range can generally overhear the RTS/Data, the CTS/ACK, or all of the *frames*[3]. For example, in the scenario shown in Figure 7, whenever node S_A transmits a packet to node R_A,

[3]The word '*packet*' implies the protocol data unit (PDU) of a higher layer whereas '*frame*' is the MAC layer PDU).

node S_B can overhear the RTS/Data frames. In the hidden-terminal scenario (Figure 4), whenever node S_A transmits a packet to node R_A, node S_B can overhear the CTS/ACK frames. In the scenario of Figure 6, where all the nodes are within one-hop, every node can overhear all the four frames. However, the above observation is not always true. For example, in the scenario-1 (Figure 2), whenever node S_A transmits a packet to node R_A, node S_B *cannot* overhear any of the four frames. This is also true for node S_A in the asymmetrical information scenarios (3). In order for the simplicity of the presentation, we will exclude these violation cases, until Section V-B. Also, we have simply assumed that the sensing range is equal to the transmission range, and we will come back to this point in VII-A.

Fig. 7. Scenario where Senders Overhear RTS/Data

A. Estimation of the number of active nodes (n)

In order to estimate the number of active nodes within the contention range, every node will maintain a list. Whenever a node overhears a frame (RTS/CTS/Data/ACK), it will insert the ID of the *sender* of the flow into the list if the ID does not exist in the list. This implies that on hearing a RTS/Data frame, the nodes should add the *source* ID into the list, whereas on hearing an CTS/ACK frame, the nodes should add the *destination* ID of the frame. In the case that the ID exists in the list, the node will simply refresh the time of the entry containing this ID. In order to prevent stale entries, an entry is deleted after a timeout interval, say W_e. With the above list, every node can easily get the number of active nodes by counting the number of IDs in its list. Now we discuss how to adaptively determine the value of W_e. It is intuitive that, to get the (current) estimate (let us say n_e) of *n*, the W_e must be at least equal to the previous estimate (let us say $n_e^{'}$) of *n*, therefore,

$$\begin{cases} W_e \geq n_e^{'} \\ n_e = \ the\ number\ of\ unique\ IDs \end{cases} \tag{4}$$

Clearly, the value of W_e affects the precision of the n_e. Sometimes, n_e is greater than *n*, and we call this as an *over-estimation*. On the other hand, the *under-estimation* refers to the case, where n_e is smaller than *n*. Both over-estimation and under-estimation lead to wastage of the medium. Specifically, when over-estimation occurs, the nodes are restrictive and thus more slots are *idle*. On the other hand, when under-estimation occurs, the nodes are aggressive and thus increasing the likelihood of *collision*. Now let us consider two situations involving the imprecise estimation. *(i)* After an active node transmits its *last* packet and thus becomes inactive, its ID will *not* be deleted by other nodes until W_e number of packets have been transmitted over the medium. This results in the over-estimation; *(ii)* When an inactive node becomes active, other nodes are *not* aware of this, until the node transmits its *first* frame. This results in the under-estimation. However, in our algorithm, whenever a node becomes active, the node will immediately enter the aggressive mode and thus transmits its first RTS immediately. Therefore, the under-estimation, if occurring, will last for a very short duration.

In order to cope with the over-estimation problem, we should use a value of W_e as *small* as possible. However, if W_e is too small, under-estimation may occur as explained below. Due to short-term unfairness, a flow may not be able to initiate any transmission for a long time though it is active. As a result, other nodes will delete the corresponding ID, leading to under-estimation. In order to cope with this dilemma, we propose *inactive-notification mechanism*. In particular, whenever a node sends out its *last* packet in its queue (i.e., switching from active to inactive), it should use a bit in the RTS/Data to tell other nodes that it is becoming inactive. Moreover, the receiver should also piggyback this notification in the *responding* CTS/ACK frames, as some nodes may not overhear the RTS/Data frames. All the nodes that hear the notification, rather than inserting or refreshing the sender's ID of this flow, should *delete* the entry that contains the sender ID from the list. It is clear that the inactive-notification mechanism greatly solves the over-estimation problem. Therefore, we can use a large W_e value to avoid the under-estimation problem. In the simulations, empirically we found that the following values of the W_e are better suited:

$$W_e = \begin{cases} 6 \times n_e^{'} & when\ n_e^{'} \leq 10 \\ 4 \times n_e^{'} & when\ n_e^{'} > 10 \end{cases} \tag{5}$$

B. Estimation of the actual share (w_x)

We now discuss how to estimate w_x, the actual proportion shared by node x during the previous window W_{cyc}. This can be done by maintaining a transmission history[4] at *each* node. Whenever a node hears a Data or ACK frame transmitted over the medium, the node will insert the *sender*'s ID of the *packet* into its history. If a node overhears both the Data and ACK frames belonging to the same handshaking, it should add the ID only *once*. By maintaining such a history, a node can easily know its actual share w_x during the latest window by simply checking how many times (say *m*) its *own* ID appears in the window W_{cyc}. Therefore,

$$Estimation(w_x) = m/n \tag{6}$$

As an example let us consider the history (beginning with the most recent entry) $\{A, A, B, C, A, C, B, A, ...\}$ where A, B, and C are the IDs of the senders. Assuming that the estimation of n from the ID list is equal to 3, then the estimations of w_x for nodes A, B, and C are 2/3, 1/3, and 0, respectively, since we only need to look at first three entries in the history, i.e, $\{A, A, B\}$. In fact, in estimating w_x at node x, the node does not need to know the exact IDs of the senders of the packets transmitted by *other* nodes. For example, when estimating w_A, node A does not need to know the IDs of the sender of the packets transmitted by nodes B and C. Obviously, a node will always know the ID of the sender of a packet if the node is the sender or receiver of the packet. The above properties are very important in the situation that

[4]It is different from the ID list maintained for the estimation of n.

the Data/ACK is not interpretable due to collisions or large sensing range as discussed in Section VII-A.

V. FMAC/CSR: Fair Medium Access Control using Cooperation between Sender and Receiver

In this section, we will introduce a novel medium contention algorithm, called FMAC/CSR, which exploits information available at the sender and receiver, and achieves MAC fairness in all the scenarios discussed in Section II.

A. Differentiated Access at the Sender

We first discuss how to detect the deviation of a node's actual share from its fair share. As an example let us consider the history (beginning with the most recent entry) $\{A, B, A, C, B, A, D, E, C, ...\}$ where A, B, C,D, and E are the IDs of the senders. Assuming that the n (estimated from the ID list) is equal to 5, then the estimated actual shares w_x for nodes A, B, C, D, and E are 2/5, 2/5, 1/5, 0, and 0, respectively, since we need only look at the most recent transmission window, i.e., $\{A, B, A, C, B\}$. As the fair share (ϕ_x) for each node should be 1/5, nodes A and B have over-used the medium during the window, while node C has had the normal-use of the medium. Nodes D and E, however, have under-used the medium.

Based on its history, a node can also measure how much it has over-used or under-used the medium. With this measurement, a node decides the *degree* by which it should be aggressive or restrictive, which is represented by N_a and N_r, respectively, and can be derived as follows. First, the node will check its actual share in the latest transmission window. If it has had normal-use of the medium in the window, N_a and N_r are set equal to zero (e.g., at node C in the above example). On the other hand, if the node has under-used the medium in the latest transmission window, N_a is initialized to *one*. In this case, the node will then check the next transmission window by sliding by one entry in the history (i.e., the window of n entries after skipping the most recent entry). If it still has under-used the medium during this window, N_a will be incremented by one. This process will continue until the node finds its share becoming normal in a transmission window. In the example given above, N_a at nodes D and E are equal to 2 and 3, respectively. Similarly, we can get the value of N_r if a node has over-used the medium during the latest transmission window. For instance, N_r at nodes A and B are equal to 3 and 2, respectively.

Once a sender has found the mode it should enter into, and the corresponding degree, it can compensate for the under or over usage using two different approaches. To explain the two approaches, let us consider a node which is in restrictive mode. The first approach is to make the node to defer its transmission until it enters the normal mode. This is a *deterministic* approach, which achieves the desirable fairness, but potentially leads to substantial capacity degrade as shown in [11]. The second approach is to assign a large contention window to the node. This is a *probabilistic* approach [7], which results in unfairness whenever the node generates a small back-off timer though the CW is large. Rather than following any of the two approaches, we use a hybrid of the two.

Now we discuss how the sender should contend for the medium whenever the medium becomes idle[5]. If the sender is in the aggressive mode, it will generate a random value from $[0, X]$ where $X = max(n, 2n - N_a)$. The X assigned in this manner is to reduce the probability of collisions when multiple senders are in aggressive mode. If a sender is in the normal mode, it will generate a random value from $[2n, CW]$, where CW is the contention window. Clearly, the nodes in the aggressive mode will get higher priority than the nodes in the normal mode. On the other hand, if the sender is in the restrictive mode, it will first defer by a time equal to $(N_r + 1) \times TxTime(packet)$ where $TxTime(packet)$ is the time needed to transmit a *packet* including overheads of RTS, CTS and ACK. Then, the node will generate a random value from $[2n, CW \times N_r]$. Whenever the back-off timer expires and the medium is idle, the node will transmit irrespective of its mode.

Note that whenever the medium becomes idle again after a busy state, the mode of the node and the corresponding degree of aggresiveness/restrictiveness are recomputed. Also, the back-off timer will be regenerated, which is completely different from the freezing mechanism used in IEEE 802.11. However, the Contention Window (CW) is manipulated as in the Binary Exponential Back-off (BEB) for its efficiency in resolving collision in the normal and restrictive mode. For example, after a collision, the CWs at the colliding nodes will be doubled, while the CW is reset after a successful transmission. The overall process followed at the senders is summarized in Figure 8.

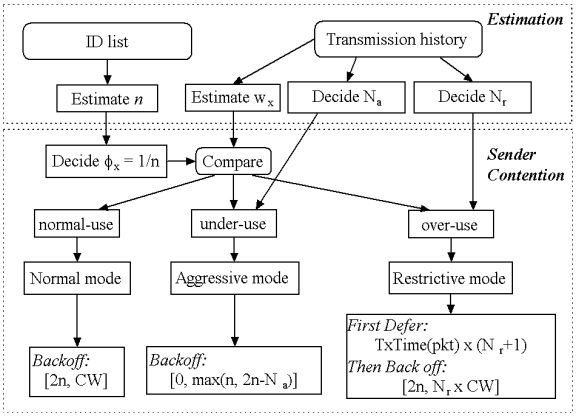

Fig. 8. Overall Process Followed at the Senders

B. Receiver Coordination: Over-use Notification

Now we address the violation case described in Section IV, where a node can hear *neither* the RTS/Data *nor* the CTS/ACK frames. For example, in the scenario-1 (Figure 2), since node S_A cannot overhear any of the frames exchanged between nodes S_B and R_B, the n_e estimated at node S_A is always equal to one. Similarly, n_e at node S_B is also always equal

[5]In fact, the contention will begin only if the medium has been idle for a duration equal to DIFS or Extended IFS (EIFS).

to one. Therefore, both nodes S_A and S_B will *always* operate in the *normal* mode, and thus, in this topology, the algorithm proposed thus far will show similar performance as that under the IEEE 802.11.

In order to cope with the above problem, we notice that the receiver node R_B can overhear the ACK frame whenever node S_A transmits a packet to node R_A. Therefore, n_e estimated at node R_B is equal to two. This is also true for the node R_A. In fact, as stated in Section II, two flows are contending with each other only if either the sender or the receiver of one flow is within the range of the sender or the receiver of the other flow. With this observation, we propose the *receiver coordination mechanism*. In particular, whenever a receiver, based on its own understanding, notices that the sender has *over-used* in the latest window, the receiver will try to slow down the sender. This can be realized through several ways. One method is that the receiver just conveys its estimated n_e to the sender. Therefore, in the above example, the sender will have two estimates of *n* and it should choose the *maximum* one, since both the sender and the receiver may under-estimate *n* in different scenarios. Using this method, n_e at nodes S_A and S_B (Figure 2) is equal to two, which reflects the real state of the contention. However, since the transmission history at node S_A will *never* contain node S_B's ID, node S_A will *always* operate in the *restrictive* mode. The same is true for S_B, leading to drastic throughput degrade.

Alternatively, to slow down the sender, the receiver can choose *not* to respond to the sender's transmission request, or responds to the request *but* can *piggyback* an *over-use notification* in the ACK frame. The pros and cons of these two methods have already been discussed in [11] in a different context, and we choose the piggyback method to regulate the sender's rate. The over-use notification should contain the restrictive degree N_r, which is calculated by checking the sender's ID in the receiver's transmission history. Whenever the sender receives an over-use notification, if it is already in the restrictive mode with a degree equal or greater than the one contained in the notification, it will simply ignore this notification. Otherwise, the sender will put itself in the restrictive mode with the degree contained in the notification.

C. Receiver Coordination: Request for RTS (RRTS)

While the over-use notification mechanism will greatly improve the fairness in scenarios like scenario-1 (Figure 2), the short-term unfairness still remains and the aggregate throughput degrades substantially. The reason is as follows. Consider that the receiver (say node R_A) notices that the sender (i.e., node S_A) has been aggressive in the past window, and thus R_A piggybacks an over-use notification. Upon receiving the notification, S_A will defer for a long enough duration enabling the node S_B to transmit a packet. However, since node S_B is unaware that node S_A is gracefully deferring and the CW at node S_B may be very large due to its futilely retrying during the previous transmission of flow A, the node S_B may not recognize this opportunity to transmit its packet. Therefore, the medium is unnecessary *idle* for a long time, and then node S_A may get control of the medium *again*, explaining the large throughput degrade and the short-term unfairness.

The main problem in the above scenario is that the sender cannot identify the contention period. In [2], the authors introduced the Request RTS (RRTS) frame at the receiver to help the sender to identify the *contention period*. However, many related issues remain unaddressed. Here we refine the mechanism. First, we discuss how a node knows that one or more senders have packets to transmit to the node itself. In particular, every node will maintain a list, which records the IDs of *active* senders. Whenever a node receives a packet that is addressed to the node itself, it will add the sender ID of the packet into the list. On the other hand, whenever an inactive-notification is piggybacked in the packet or an ID has been in the list for too long without refreshing, it will delete the ID.

Now we discuss how the RRTS mechanism should be used. Whenever the medium becomes idle, and if a receiver (e.g., node R_B) notices that one of its *active* senders should be in the aggressive mode, the receiver will send out an RRTS to the sender (i.e., node S_B). Similar to the over-use notification, the RRTS frame should contain the aggressive degree N_a, which is calculated by checking the sender's ID in the receiver's transmission history. In the case there are multiple senders that should be in the aggressive mode, the receiver will send out an RRTS to the node who has the largest aggressive degree. Upon receiving the RRTS, if the sender (i.e., node S_B) is in the restrictive mode, it will ignore the RRTS. Otherwise, it will contend for the medium as per the aggressive mode and with the degree contained in the RRTS.

Two issues need further investigation. *First*, we should try to avoid the collision of RRTS frames as there may be more than one receivers intending to send out an RRTS frame at the same time. *Second*, while the over-use notification will *not* introduce much overhead as it is piggybacked in the ACK frame, the *explicit* transmission of an RRTS frame is a wastage of bandwidth. Therefore, we should try to restrict the transmission of RRTS whenever it is unnecessary. In particular, if the sender is already aware that it should be aggressive and it can identify the contention period, its receiver should not send out an RRTS frame. Let us consider the scenario presented in Figure 9, which includes two more flows (i.e., flows C and D) compared to scenario-1 (Figure 2). The senders and receivers of the two additional flows are within the range of each other. Moreover, they are within the range of R_A and R_B. Therefore, in such a scenario, generally the senders (i.e., S_C and S_D) can obtain the same information of the contention as the receivers (i.e., R_C and R_D) do, and thus the receivers should not send out an RRTS to their corresponding senders when the flows have under-used the medium. On the other hand, if flow A or B has under-used the meidum, the RRTS mechanism should be used by R_A and R_B. Moreover, if there is another contending flow that is in normal/restrictive mode, the RRTS should be sent out before the sender of that flow transmits its RTS. In summary, we need to investigate some mechanisms, which can intelligently decide whether or not an RRTS frame should be sent, and if it is to be sent, when.

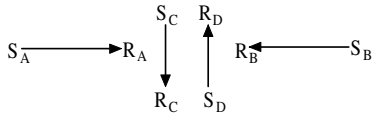

Fig. 9. Scenario Illustrating whether or not RRTS needed

To cope with the first issue discussed above, before a receiver sends out an RRTS, it will first back off by a random value. To cope with the second issue, the priority for the medium access should be in the following order: first, senders in the aggressive mode; second, receivers in the aggressive mode; third, senders in the normal mode; and the last, senders in the restrictive mode. To give priority to the senders that are operating in the aggressive mode, the random value for RRTS can be within the range $[2n, Y]$ where $Y = max(3n, 4n - N_a)$. If the medium is still idle after backing-off, the receiver will send out an RRTS frame. All the other nodes that overhear the RRTS will defer their transmission by a duration, which is indicated in the RRTS frame and is equal to $(DIFS + 2n \times Slot + TxTime(RTS))$. To give priority for the RRTS, all the senders that are operating in normal/restrictive mode will generate a random back-off timer which is at least $4n$ rather than $2n$ as used in Section V-A. Similarly, the minimum value of the back-off timer for the nodes in the restrictive mode should change from $2n$ to $4n$. In this way, we avoid the unnecessary transmission of RRTS but do not compromise on the effectiveness of the RRTS mechanism.

Summary of FMAC/CSR

The basic idea of FMAC/CSR is to make both the sender and the receiver of a flow to contend for the medium or restrict the usage of the medium in a cooperative manner. However, in our FMAC/CSR, the sender still takes a more active role that the receiver does. In Table I, we summarize the FMAC/CSR. In the table, FMAC/CSR-1 refers to the FMAC/CSR that only adopts the differentiated access at the senders, while FMAC/CSR-2 refers to FMAC/CSR-1 plus the over-use notification mechanism. Obviously, FMAC/CSR-3 refers to the FMAC/CSR-2 plus the RRTS mechanism.

TABLE I

SUMMARY OF THREE CATEGORIES OF FMAC/CSR

Mode	Side	FMAC/CSR-1	FMAC/CSR-2	FMAC/CSR-3
Aggressive	Sender	*Back off:* [0, max(n, 2n-N_a)]	*Back off:* [0, max(n, 2n-N_a)]	*Back off:* [0, max(n, 2n-N_a)]
	Receiver	NA	NA	*First Back off:* [2n, max(3n, 4n-N_a)] *Then send RRTS*
Normal	Sender	*Back off:* [2n, CW]	*Back off:* [2n, CW]	*Back off:* [4n, CW]
	Receiver	NA	NA	NA
Restrictive	Sender	*First Defer:* TxTime(pkt) x (N_r + 1) *Then Back off:* [2n, N_r x CW]	*First Defer:* TxTime(pkt) x (N_r + 1) *Then Back off:* [2n, N_r x CW]	*First Defer:* TxTime(pkt) x (N_r + 1) *Then Back off:* [4n, N_r x CW]
	Receiver	NA	*Piggyback:* Over-use notification	*Piggyback:* Over-use notification

VI. SIMULATION RESULTS

In this section, we present the simulation results to compare our proposed FMAC/CSR algorithm with the IEEE 802.11. The simulations were performed under the NS-2 with CMU wireless extensions [6]. All the flows are single-hop and each of them uses a Constant Bit Rate (CBR) traffic generating 200 packets per second. Each packet is 1000-bytes long, resulting in a traffic source rate of 1.6 Mbps. The raw bandwidth is set to 2 Mbps, leading to a *maximum* throughput of about 1.4 Mbps due to the overheads of IEEE 802.11. The source rate is made greater than the medium capacity to ensure that the contending nodes always have packets to send. The static routing is used. The sensing range is equal to the transmission range. Mobility, capture and wireless errors are not considered in the simulation. The simulation time is 200 seconds.

The well-known Jain's index [13] is used as the main measure, which is defined as follows:

$$F_J = (\sum_{i=1}^{N} \gamma_i)^2 / (N \sum_{i=1}^{N} \gamma_i^2) \quad (7)$$

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.

A. Hidden-terminal topologies

The Jain's index for the hidden-terminal topologies (Figure 4) is presented in Figure 10. For the IEEE 802.11, when *w* is small (e.g., 2), the index is very small (about 0.52) compared to the absolute fairness (i.e., unity), implying substantial short-term unfairness. On the other hand, when *w* is very large, the index is close to unity (though not shown in the figure due to the constraint of the figure size), implying the long-term fairness. Compared to the IEEE 802.11, the FMAC/CSR-1 greatly improves the fairness as shown in the figure. However, in this topology, the FMAC/CSR-2 and FMAC/CSR-3, which incorporate the receiver-coordination mechanism(s), do not show advantage in comparison to the FMAC/CSR-1. The reason is that the senders in this topology can get the complete information about the contending flows, and thus the receiver coordination mechanisms do not play any roles.

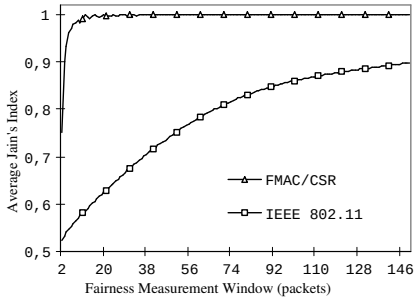

Fig. 10. Fairness Index in Hidden-terminal Topologies

As clearly pointed out in [14], generally, *maximizing capacity* and *achieving fairness* are two *conflicting* objectives

in wireless ad hoc networks. Therefore, when evaluating a fairness algorithm, it is also necessary to look at the throughput results to ensure that the algorithm does not degrade the throughput too much. Table II presents the throughputs under different schemes. While the FMAC/CSR greatly improves the fairness, it also improves the aggregate throughputs. This shows the advantage of our FMAC/CSR. Note that the aggregate throughput under FMAC/CSR-3 degrades insignificantly compared to those of FMAC/CSR-1 and FMAC/CSR-2 due to the reason that under normal and restrictive modes in FMAC/CSR-3, the minimum back-off time at the sender is $4n$ rather than $2n$ (see Table I). Now we explain why the throughput under FMAC/CSR is greater than that under IEEE 802.11. As discussed in Section II-B, under IEEE 802.11, whenever the two senders contend for the medium, their RTSs are very likely to collide, leading to wastage of bandwidth. On the contrary, under the FMAC/CSR, since the senders can cooperate in a distributed manner (e.g., when one sender is in the restrictive mode, then the other one is in the aggressive mode), the chances of collisions and the medium being idle are greatly reduced, explaining the throughput improvements.

TABLE II

THROUGHPUT UNDER HIDDEN-TERMINAL TOPOLOGIES

Throughput (Mbps)		IEEE 802.11	FMAC/CSR-1	FMAC/CSR-2	FMAC/CSR-3
Average	S_A to R_A	0.678	0.720	0.720	0.717
	S_B to R_B	0.676	0.720	0.720	0.717
Aggregate		1.354	1.440	1.440	1.434

B. Asymmetrical Information Topologies

As long-term unfairness is known to occur in the asymmetrical information topologies, we first present the throughput results in Table III. It is easy to see that, under IEEE 802.11, flow B gets almost entire bandwidth in comparison to flow A. The reason of long-term unfairness has been explained in Section II-B. Under FMAC/CSR-1, the fairness substantially improves, however, it does not achieve long-term fairness completely. Moreover, the aggregate throughput degrades. The reason is as follows. As the n estimated at node S_A is equal to one, S_A always operates in the normal mode. If node S_B operates in the aggressive/normal mode, as discussed in Section II-B, S_B can always contend efficiently while S_A will retry futilely, resulting in a large CW. After transmitting two packets consecutively, S_B will enter the restrictive mode and thus defer its transmission for a long time. However, S_A may still back off since its CW is very large. Therefore, the medium is idle for a substantial time before any transmission occurs, explaining the degradation in the aggregate throughput. Moreover, if S_A's back-off timer is very large (e.g, when the CW is equal to 1023), S_B will begin to transmit again after deferring and backing off, explaining the slight long-term unfairness remaining in FMAC/CSR-1. In summary, node S_A's futile retries during flow B's transmission is the underlying bane of the performance degradation.

The fairness as well as the aggregate throughput substantially improves under FMAC/CSR-2, which incorporates the over-use notification mechanism. However, the performance does not improve any further under FMAC/CSR-3, which includes RRTS mechanism. To explain why FMAC/CSR-2 improves the performance compared to FMAC/CSR-1, consider that, after node S_B transmits two packets consecutively, it enters the restrictive mode and defers its transmission by a time needed for transmitting two packets. Therefore, once node S_A gets control of the medium, it can also transmit two packets consecutively after which node S_B will begin to contend for the medium. If the over-use notification mechanism is used (as in FMAC/CSR-2), node S_A will defer. On the contrary, without the over-use notification (as in FMAC/CSR-1), S_A will futilely retry as discussed above. In other words, the over-use notification mechanism helps node S_A to identify the contention period more accurately rather than futilely retrying, explaining the performance improvements in FMAC/CSR-2. In this manner, the over-use notification mechanism has accomplished the task (i.e., helping S_A to identify the contention period) that the RRTS mechanism is supposed to do, and therefore the RRTS mechanism introduced in FMAC/CSR-3 does not show any additional advantage. Again, the aggregate throughput under FMAC/CSR-3 degrades slightly compared to that of FMAC/CSR-2 due to the reason that under FMAC/CSR-3, the minimum back-off time at the sender is $4n$ rather than $2n$ (see Table I).

The Jain's index is presented in Figure 11. FMAC/CSR-2 achieves short-term fairness, and thus automatically ensures long-term fairness. Since FMAC/CSR-2 and FMAC/CSR-3 show similar fairness, we do not present the index under FMAC/CSR-3 to avoid the cluttering of the graph.

TABLE III

THROUGHPUT UNDER ASYMMETRICAL INFORMATION TOPOLOGIES

Throughput (Mbps)		IEEE 802.11	FMAC/CSR-1	FMAC/CSR-2	FMAC/CSR-3
Average	S_A to R_A	0.073	0.538	0.718	0.716
	S_B to R_B	1.345	0.628	0.718	0.716
Aggregate		1.418	1.166	1.436	1.432

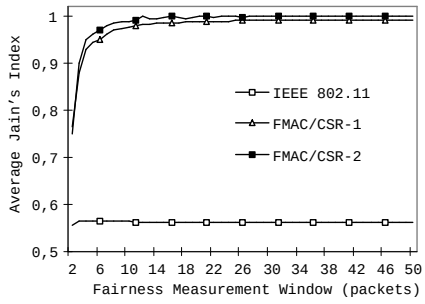

Fig. 11. Fairness Index in Asymmetrical Information Topologies

C. Scenario-1 of Figure 2

In this scenario, under IEEE 802.11, each of the two flows gets only about 0.29 Mbps, which is much less than the expected 0.7 Mbps. The reason is as follows. Consider the situation that node S_A sends out an RTS to node R_A, and then node R_A sends back the CTS. As node S_B is unaware

of this CTS, it may send out an RTS, leading to a collision at node R_B. A node (e.g., node R_B) that detects a collision will defer for the Extended Inter-Frame Space (EIFS) duration, which is much smaller than the time needed for the complete transmission of a Data frame. Since the RTS/CTS handshaking is successful for node S_A, it will transmit the Data frame. During the transmission, S_B may initiate its RTS again. If the RTS arrives at node R_B after the EIFS deferment is over, R_B will respond with CTS, which will collide with the Data frame from node S_A to R_A, and thus node R_A will *discard* the Data frame and defer by an EIFS. Now, for node S_B, since the RTS/CTS is successful, it transmits the Data frame. The Data frame is likely to be destroyed by node R_A's CTS in a similar manner, resulting in substantial bandwidth wastage. The dual busy tone multiple access (DBTMA) proposed in [9] can solve this problem, but two additional busy-tone channels are required. For simplicity, here we make a slight modification in the IEEE 802.11, that is, whenever a node detects a collision, rather than deferring by an standard EIFS duration, it will defer for a large enough duration enabling transmission of a Data frame. We call this as the *Large-Col-EIFS* mechanism. In this subsection, we present the results assuming that this mechanism is included. Though the pros and cons of the *Large-Col-EIFS* need further investigation, it is also used in the scenario of Figure 15 as the scenario also contains the case that two senders are three hops away.

The aggregate throughput is presented in Table IV and the Jain's index is presented in Figure 12. We notice that the aggregate throughput in IEEE 802.11 with *Large-Col-EIFS* mechanism greatly improves. As for the fairness, IEEE 802.11 exhibits substantial short-term unfairness. Moreover, the FMAC/CSR-1 does not improve the fairness or throughput. On the contrary, the FMAC/CSR-2, which incorporates the over-use notification mechanism, greatly improves the fairness but at the cost of aggregate throughput. The FMAC/CSR-3, which includes the RRTS mechanism, improves the short-term fairness as well as the aggregate throughput compared to FMAC/CSR-2. All the above observations have been clearly explained in subsections V-B and V-C.

TABLE IV

THROUGHPUT UNDER TOPOLOGY OF FIGURE 2

Throughput (Mbps)		IEEE 802.11	FMAC/CSR-1	FMAC/CSR-2	FMAC/CSR-3
Average	S_A to R_A	0.703	0.701	0.532	0.582
	S_B to R_B	0.707	0.700	0.530	0.582
Aggregate		1.410	1.401	1.062	1.164

D. Scenario-7 of Figure 5

The throughput results are presented Table V and the Jain's index is presented in Figure 13. It is easy to see that, under IEEE 802.11, flow B gets much more bandwidth than flow A, due to the imprecise collision detection as explained in Section II-C. Under FMAC/CSR-1, the fairness substantially improves. Again, the FMAC/CSR-2 and FMAC/CSR-3, which incorporate the receiver-coordination mechanisms, do not show advantage in comparison to the FMAC/CSR-1 as the senders can get the complete information about the contending flows.

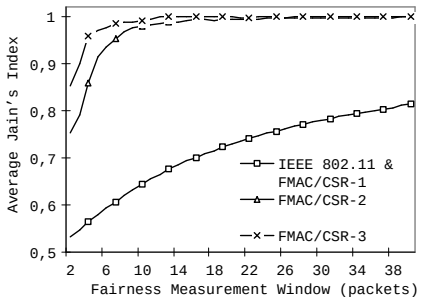

Fig. 12. Fairness Index under Topology of Figure 2

TABLE V

THROUGHPUT UNDER SCENARIO-7 (FIGURE 5)

Throughput (Mbps)		IEEE 802.11	FMAC/CSR-1	FMAC/CSR-2	FMAC/CSR-3
Average	S_A to R_A	0.672	0.720	0.720	0.717
	S_B to R_B	0.766	0.720	0.720	0.717
Aggregate		1.438	1.440	1.440	1.434

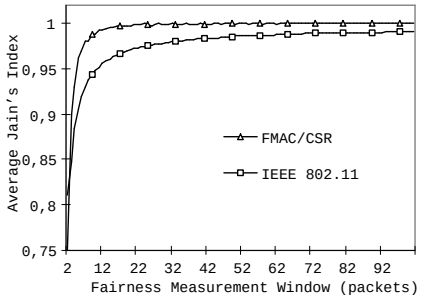

Fig. 13. Fairness Index under Scenario-7 (Figure 5)

E. High-contention topology

In this subsection, we present the results for the high-contention scenario shown in Figure 6. The Jain's index is displayed in Figure 14 and the throughput results are presented in Table VI. The IEEE 802.11 exhibits substantial short-term unfairness. On the contrary, the FMAC/CSR-1 greatly improves the short-term fairness compared to the IEEE 802.11, and again the receiver coordination mechanisms do not play any role here. The aggregate throughput under FMAC/CSR degrades slightly compared to that under IEEE 802.11.

Similar fairness benefits were observed when the number of contending nodes is *larger* than five but the nodes were within the one hop distance.

TABLE VI

THROUGHPUT IN HIGH-CONTENTION TOPOLOGY

Throughput (Mbps)	IEEE 802.11	FMAC/CSR-1	FMAC/CSR-2	FMAC/CSR-3
Aggregate	1.410	1.405	1.405	1.370

F. A complex topology

In this subsection, we consider a complex topology shown in Figure 15, which combines the main scenarios discussed so far. For convenience of explaining the relative positions of

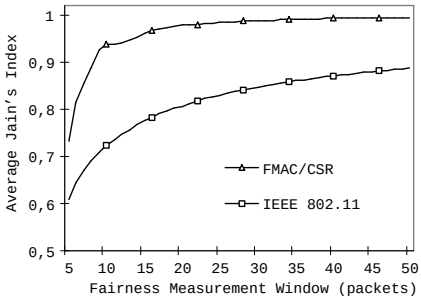

Fig. 14. Fairness Index in High-Contention Topology

the nodes, we have drawn two concentric circles in the figure. The distance between the sender and receiver of each flow is 200 meters, except for the flow from node 16 to 17, where the distance is set in a way such that the two nodes are in the ranges of the nodes on the inner circle but out of the ranges of the nodes on the outer circle. The diameter of the inner circle is 200 meters. Therefore, the diameter of the outer circle is 600 meters. The angle between any two neighboring flows is 45 degrees, except for the flow from node 16 to 17. The *Large-Col-EIFS* mechanism discussed before is adopted in this topology.

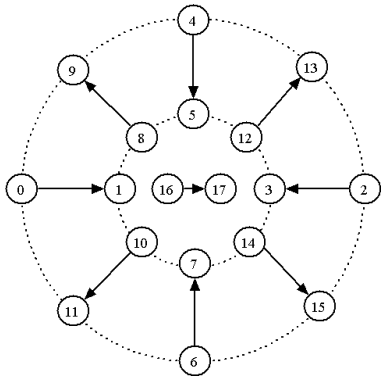

Fig. 15. Scenario-9: Complex Topology

The throughput results are presented in tables VII and VIII, while the Jain's index is displayed in Figure 16. In Table VII, the flow ID is the same as the node ID of the flow's sender. Under IEEE 802.11, four flows (i.e., flows from node 0 to 1, node 2 to 3, node 4 to 5, and node 6 to 7) starve, while the flow from node 16 to 17 gets much higher throughout than the remaining flows. On the other hand, the bandwidth under FMAC/CSR-3 is distributed quite evenly among the flows. From Table VIII, we notice that the aggregate throughputs under all three FMAC/CSR schemes improve compared to that under IEEE 802.11, showing the merits of our FMAC/CSR in complex scenarios. The FMAC/CSR schemes also achieve short-term fairness as clearly shown by the Jain's Index results in Figure 16.

G. Verification of Estimation Algorithm

So far, using Constant Bit Rate (CBR) traffic, we have seen that our FMAC/CSR substantially improve the fairness. From the simulation traces, we also found that the estimation of *n*

TABLE VII

THROUGHPUTS IN COMPLEX TOPOLOGY (FIGURE 15)

Flow ID	0	2	4	6	8	10	12	14	16
IEEE 802.11	0.000	0.000	0.000	0.000	0.156	0.193	0.195	0.134	0.509
FMAC/CSR-3	0.149	0.146	0.149	0.143	0.128	0.128	0.128	0.121	0.153

TABLE VIII

AGGREGATE THROUGHPUTS IN COMPLEX TOPOLOGY

Throughput (M bps)	IEEE 802.11	FMAC/CSR-1	FMAC/CSR-2	FMAC/CSR-3
Aggregate	1.187	1.258	1.210	1.244

is quite precise. In this section, we evaluate the precision of the estimate of *n* when the traffic is dynamically changing.

Exponential On/Off traffic is used at the nodes. The average "on" time is 0.3 s, whereas the average "off" time is 0.7 s. In the "on" state, CBR traffic is generated as explained earlier. The topology depicted in Figure 6 is used for the study. We dynamically record n_e and *n* throughout the simulation time. The interval between two consecutive samples is 0.005 s, which is very close to the time needed for the transmission of a packet. Figure 17 presents the results during a typical duration of 2 seconds. We notice that n_e is equal to *n* most of the time.

VII. DISCUSSION AND RELATED WORK

A. Further Comments on FMAC/CSR

FMAC/CSR when Sensing Range > Transmission Range: In FMAC/CSR, a node needs to know the number of active nodes (i.e., n) within the contention range and its actual share (i..e., w_x), which can only be estimated in a distributed manner in wireless ad hoc networks. In Section IV, we have proposed an estimation algorithm by simply assuming that the transmission range (TR) is equal to the sensing range (SR). In practice, the SR may be greater than the TR. Therefore, our estimation algorithm, as well as other algorithms ([7], [11], [12]) based on overhearing (including the Virtual Carrier Sensing mechanism in IEEE 802.11) may not work in such a scenario. However, we should note that the unfairness problem (discussed in Section II), the Fairness Framework (in Section III), and the FMAC/CSR itself (in Section V) are quite general and also applicable when SR>TR.

Now we discuss how to modify our estimation algorithm when SR>TR. First of all, we should note that the transmission range of RTS/CTS frames is much greater than that of Data/ACK frames since the length of the control frames (RTS/CTS) is very short compared to that of Data frame, and since the control frames are always transmitted at the lowest rate. This has also been pointed out in [1]. On the contrary, the sensing range for all the frames (RTS/CTS/Data/ACK) should be the same since the range only depends on the carrier sensing energy threshold. Therefore, if the threshold is set to an energy level such that the control frames (RTS/CTS) are always interpretable, and hence the transmission range of the RTS/CTS frames is equal to the common sensing range. In fact, as clearly shown in [17], this is true for the emerging standards IEEE 802.11a/h. In such a situation, whenever a

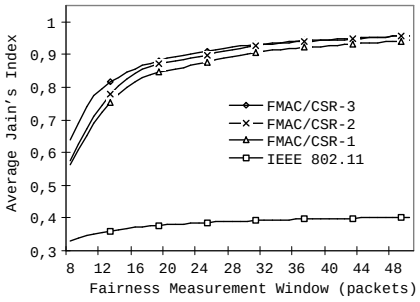

Fig. 16. Fairness Index in Complex Topology (Figure 15)

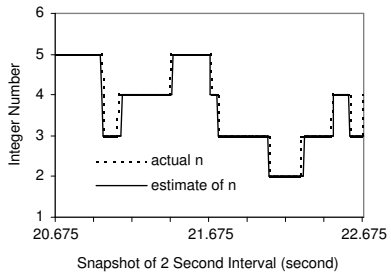

Fig. 17. Comparison between Actual and Estimation of *n*

packet is transmitted over the medium, a node within the sensing range will always overhear the RTS/CTS clearly. Therefore, if an algorithm assumes SR=TR but only relies on the overhearing of RTS/CTS frames, it can work even when SR is greater than the TR for the Data frames. This is true in our estimation algorithm for the number of active nodes *n* since it only relies on the overhearing of RTS/CTS. Therefore, when SR>TR, we only need to make a slight modification in the estimation of the actual share w_x. Specifically, a node should add an invalid ID (e.g., -1) whenever it detects a Data/ACK frame but it cannot interpret the contents of the frame[6]. To cope with the situation that a node can detect the Data and ACK frames both of which are interpretable but belonging to a same handshake, the node can start a timer on detecting a Data frame, and will not add '-1' ID into its history *another* time if an ACK frame is detected before the timer expires. Moreover, we can further prevent the wrong inserting of '-1' ID by maintaining timers when RTS/CTS is clearly overheard. As we have mentioned before, a node under our FMAC/CSR needs only to know the actual share of itself. Moreover, the estimation of w_x at node x does not need to know the exact IDs of the senders of the packets transmitted by other nodes. Therefore, the modified estimation algorithm for actual share can work well when the SR is greater than the TR of Data frames. The above discussion also applies to the situation that a collision is detected since in this case also the frame contents are not interpretable.

In the case that the sensing range is even greater than the transmission range of RTS/CTS (e.g., under IEEE 802.11b), the above modified estimation algorithm for the actual share will still work, though the estimation of n may not always be precise. However, with our proposed receiver coordination mechanism, any ill effects due to wrong estimation will occur only when both the sender and receiver have wrong estimation of n due to a very large sensing range. Alternatively, we can get an estimate of *n* using some heuristics method as done in [3], [4] and [5].

[6]In such a situation, the type (Data or ACK) of a frame can be identified based on the duration for which the medium is busy due to the transmission of the frame.

FMAC/CSR under Spatial Reuse: In this paper, we have shown that FMAC/CSR can achieve desirable fairness in the scenarios where all flows are contending with each other. However, in a multi-hop wireless ad hoc network, it is common that two flows are not contending with each other (and thus they can transmit simultaneously) but any transmission of a flow has a global effect in the whole network as clearly shown in [15]. Therefore, to achieve fairness when spatial reuse is possible, our FMAC/CSR may need further investigation. However, in such a scenario, the main modification needed is to refine our fairness model (discussed in Section III-A) such that it can reflect not only the local contention but also the global constraints in achieving fairness. Once the fair share of a flow has been determined, our FMAC/CSR that exploit local cooperation can be adopted to achieve fairness in every neighborhood.

FMAC/CSR under Variable Packet Length and Different Flow Priorities: As done in many research-work, in this paper, we have assumed that the packets have the same length and the applications have the same priority, which are practical assumptions in wireless networks [15]. However, our FMAC/CSR can be easily extended for the scenarios with variable packet length and different flow priorities. Again, the main modification needed is to refine the fairness model to reflect such a case.

B. Related work

The fairness problem in the random wireless MAC protocols was first highlighted in [2]. Recently, several researchers have proposed a scheduler that is *overlaid* on the top of the MAC layer, to address the unfairness problem due to the *location-dependent contention* in multi-hop wireless ad hoc networks (e.g., [14], [15]). However, to implement the fair scheduling algorithm, a scheduler needs global information and synchronization, which are difficult to obtain in the multi-hop wireless ad hoc networks. On the contrary, some other works (e.g., [7], [11], [16], [19]) aim to achieve fairness at the MAC layer itself. Our work belongs to this category. In [16], to achieve proportional fairness, the authors have developed a distributed p-persistent MAC protocol, which dynamically adjusts the probability with which a sender should access the medium based on its observation of the medium states (e.g., collision, idle, or busy). However, as shown in our work, based on the information available at a sender only, the estimation at the sender is not precise and thus it is difficult to achieve fairness. The authors in [19] try to emulate the Self-Clocked Fair Queueing [8] in wireless ad hoc networks, which however needs a synchronized virtual time. Similar to our work, the authors of [7] and [11] try to make the medium access more fair based on overhearing in the neighborhood.

For example, in [7], they dynamically estimate the sharing of the medium and then tune the contention window (CW). Our scheme is different from theirs in many aspects. For example, as they do not estimate *n*, the ϕ_x is always fixed at *0.5* irrespective of the number of active nodes, which leads to more collisions when *n* is large. More importantly, they have not proposed any schemes similar to our receiver-coordination mechanisms, which are essential to cope with the concealed information problem. On the other hand, the authors in [11] have only focused on the asymmetrical information topologies and proposed an ordered packet scheduling algorithm, which implements a FIFO queue among the contending nodes. However, their scheme results in substantial throughput degrade as a fully deterministic method is adopted. Moreover, as shown in [18], due to the deterministic nature, deadlock is likely happen in a complex scenario. Our FMAC/CSR is quite unique as we adopt a hybrid method, which achieves fairness without unduly degrading the throughput.

VIII. Conclusions

In this paper, we have proposed a novel medium access control protocol, called FMAC/CSR, which exploits information available at the sender and receiver to achieves MAC fairness. Our FMAC/CSR is based on the dynamic estimation of the medium state. The simulation results show that the FMAC/CSR greatly improves the fairness without unduly degrading the capacity utilization. Our FMAC/CSR is novel in the sense that it is *simple*, *easy to implement*, *fully distributed*, and *adaptive*. Our main contributions include: *(i)* identification of reasons that lead to MAC unfairness, e.g., concealed information problem, imprecise collision detection mechanism, and high contention; *(ii)* definition of a general MAC fairness framework; *(iii)* proposal of a simple but efficient algorithm, which can dynamically estimate the number of *active* nodes within the contention range as well as the actual share of a node; *(iv)* proposal of the FMAC/CSR, which incorporates a suit of mechanisms, such as differentiated access at the senders, over-use notification, and Request RTS (RRTS).

References

[1] G. Anastasi, E. Borgia, M. Conti, E. Gregori, "IEEE 802.11 Ad Hoc Networks: Performance Measurements", Proc. Workshop on Mobile and Wireless Networks (MWN 2003), 19 May, 2003.

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

[3] G. Bianchi, I. Tinnirello, "Kalman Filter Estimation of the Number of Competing Terminals in an IEEE 802.11 network," IEEE Infocom, 2003.

[4] F. Cali, M. Conti, E. Gregori, "Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit," IEEE JSAC, December 2000, pp.785-799

[5] F. Cali, M. Conti, E. Gregori, "IEEE 802.11 Protocol: Design and Performance Evaluation of an Adaptive Backoff Mechanism," IEEE JSAC, September 2000, pp.1774-1786

[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 Backoff Algorithm for IEEE 802.11 DFWMAC," ACM MOBIHOC, 2002.

[8] S. Golestani, "A Self-clocked Fair Queueing Scheme for Broadband Apllicaiotns," IEEE Infocom, 1994.

[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] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, E, Knightly, "Distributed multi-hop scheduling and medium access with delay and throughput constrains," in ACM MOBICOM, 2001.

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

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

[15] H. Lou, P. Medvedev, J. Cheng, S. Lu, "A Self-Coordinating Approach to Distributed Fair Queueing in Ad Hoc Wireless Networks," IEEE Infocom, 2001.

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

[17] D.J. Qiao, S. Chio, A. Jain, K. G. Shin, "MiSer: An Optimal Low-Energy Transmission Strategy for IEEE 802.11a/h," ACM Mobicom, 2003

[18] K. To, Y.Z. Chen, "A-DWOP: an Extension to DWOP for Multi-hop Wireless Ad Hoc Networks," available at http://www.owlnet.rice.edu/ takhoa/courses/537/

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

[20] S. Xu, T. Saadawi, "Does the IEEE 802.11 MAC Protocol Work Well in Multihop Wireless Ad Hoc Networks?," IEEE Communications Magazine, pages 130-137, June 2001