FMAC/CSR: a Fair MAC Protocol for Wireless Ad-hoc Networks

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

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

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

***Abstract*— Recently, the issue of achieving MAC layer fairness in wireless ad-hoc networks has been extensively addressed. Most of the published schemes are sender-based, meaning that a sender (of a flow) contends for the shared medium based on its own understanding of the contention on the medium. However, as clearly shown in this paper, in a wireless ad-hoc network, a sender may not have precise information of the contention on the medium, and thus a purely sender-based approach cannot always achieve fairness and sometimes it may even degrade the performance due to the incorrect understanding. On the other hand, the receiver side (of a flow) may have some crucial information that is unknown to the sender. With this observation, we propose a novel *Fair MAC* protocol, called FMAC/CSR, which achieves fairness using the *Cooperation* between the *Sender* and *Receiver*. Extensive simulation results show that the proposed FMAC/CSR substantially improves the MAC fairness without unduly degrading the throughput.**

Compared to the large number of published papers, our FMAC/CSR is unique as it achieves fairness in all the scenarios where spatial reuse is not possible. In addition, we have addressed the unfairness issue in a very general manner. Specifically, we have systematically identified the reasons of unfairness in IEEE 802.11, and proposed a framework to achieve fairness. We have also considered the practical side of FMAC/CSR when the sensing range is greater than the transmission range.

I. INTRODUCTION

Recently, wireless ad-hoc networks have attracted considerable research interest. As IEEE 802.11 [11] 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: Point Coordination Function (PCF) and Distributed Coordination Function (DCF). DCF is a Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA)-based MAC protocol, and it is popularly used in wireless ad-hoc networks. However, DCF suffers from the well-known unfairness problem.

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 [14]. 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).

In the literature, the fairness issue in wireless ad-hoc networks has been addressed in two different manners. One is to design a scheduler (e.g., [15]), which is overlaid on the top of the MAC layer, to achieve a global fairness in a *multi-hop* wireless ad-hoc network. Obviously, this approach requires some global information of the network (e.g., topology). The other manner is to address the fairness issue at the MAC layer itself (e.g., [17]), which operates in a distributed manner but may only achieve a local fairness in a neighborhood. In fact, the fair scheduling in a multi-hop network and the fair medium access in a neighborhood are somewhat orthogonal issues and may need to be addressed separately. In this paper, we address the issue of fairness at the MAC layer. Most of the published schemes in this category are sender-based, meaning that a sender (of a flow) contends for the shared medium based on its own understanding of the contention on the medium. However, as clearly shown in this paper, the senders of the flows that contend for the same medium may not be in the hearing range of each other, and thus the information obtained at the senders may not reflect the medium contention precisely. We call it *concealed information problem*, since the information needed for the fair contention is *concealed* from the senders. Our results show that the concealed information problem results in substantial unfairness in IEEE 802.11 as it is a sender-centric protocol. Moreover, due to this problem, any enhanced fair scheme (e.g., [7]) that is purely based on the information at the sender may not achieve fairness or sometimes it may even degrade the performance due to the wrong decisions made after the incorrect understanding of the contention. In addition to the concealed information problem, we have identified two other causes of unfairness in IEEE 802.11: imprecise collision detection and high contention.

To achieve fairness, we propose a general MAC fairness framework, which includes three components: a fairness model to define fair shares, a compensation model to compensate for the over-use and under-use with respect to the fair shares, and a distributed algorithm to realize the above two models. Clearly, the last component is the key to achieve fairness due to the distributed nature of the ad-hoc networks. The distributed algorithm proposed in this paper is called FMAC/CSR, which achieves fairness using the *Cooperation* between the *Sender* and *Receiver*. In FMAC/CSR, the sender as well as the receiver of a flow (say, x), in a *distributed* manner, estimate the number of *active* flows (say *n*) within the contention range, and the actual bandwidth share (say w_x) received by the flow. The estimate of *n* is a good indication of the contention degree, and therefore, it can be used by the flow 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). Based on the deviation, a flow detects the unfairness, and thus determines which mode it should enter of the following three modes, *aggressive*, *restrictive*, and *normal*, indicating how the sender and receiver of the flow should behave when contending for the medium. Specifically, according to the mode a flow enters into, the sender contends for the medium with different priorities. On the other hand, the receiver affects the sender's behavior in the contention through either implicit or explicit feedback to the sender. In FMAC/CSR, an implicit feedback, called *restrictive-notification*, is used by the receiver to slow down the sender whenever it finds (based on its own understanding) that the flow should enter the restrictive mode. On the other hand, whenever the receiver finds that the flow should enter the aggressive mode and now it is the right time for the sender to contend for the medium, an explicit feedback, called *aggressive-notification*, may be transmitted so as to expedite the sender's contention for the medium. However, when the receiver thinks that the flow has got a fair share of the medium (and thus the flow should enter the normal mode), no special action is taken by the receiver. Clearly, in FMAC/CSR, the sender of a flow even now plays a more active role than the receiver does.

In the design of FMAC/CSR, two issues need very careful consideration. *(i)* Due to the concealed information problem, the decision (e.g., the mode that a flow should enter) made by the sender may be different from that made by the receiver since they make the decision based on their own understanding. Therefore, rules must be defined that the sender and receiver should follow whenever such a discrepancy occurs. *(ii)* Benefits of cooperation among flows may be undermined due to the incorrect information at the nodes, and therefore, the events such as medium being idle, collisions, unnecessary explicit feedback, and deadlock that result in throughput degradation need much more careful consideration.

Extensive simulation results show that the FMAC/CSR achieves fairness in all the scenarios where spatial-reuse is not possible. Meanwhile, it does not unduly degrade the throughput, and it even improves the throughput in certain scenarios. The above properties make our FMAC/CSR unique compared to the large number of published papers on MAC fairness.

The remainder of the paper is organized as follows. In Section II, we describe the basic techniques of IEEE 802.11. The reasons of unfairness are systematically identified in Section III. In Section IV, we present a general MAC fairness framework. A distributed estimation algorithm and the FMAC/CSR are presented in Sections V and VI, respectively. The simulation results are presented in Section VII. Comments on FMAC/CSR and a thorough literature review are given in Section VIII. The paper is concluded in Section IX.

II. 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, a sequence of Request To Send (RTS), Clear To Send (CTS), Data, and ACK frames, is transmitted for the transmission of every single data packet. 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.

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

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. Clearly, 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.

III. Unfairness in IEEE 802.11

Though there has a substantial research-work on MAC fairness, the reasons of unfairness have not been identified very clearly. In this section, we will systematically identify the reasons of unfairness in IEEE 802.11. As our focus is to study the fairness issue at the *MAC layer*, we only consider the scenarios in which spatial reuse is not possible, i.e., all the flows in the network are contending with each other. In Section VIII-A, we will come back to this point. Similar to most of the published work on MAC fairness, mobility, wireless error, and signal capture are not considered in this paper. Also, we consider only the single-hop flows. For each flow, a Constant

Bit Rate (CBR) traffic is adopted and the traffic rate is large enough for the flow to occupy the entire channel capacity as unfairness occurs only when the system is over-loaded. The above assumptions are adopted to clearly bring out the problems that are discussed below, though they may also affect MAC fairness.

A. 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. In this subsection, we try to identify all the possible scenarios, which have two contending flows and exhibit short-term or long-term unfairness. Obviously, the discussion applies when more than two flows are involved.

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 first assume that S_A, R_A, S_B, R_B are different nodes. Based on whether S_A is in the range of S_B and R_B, we present the relationship between node S_A and *flow* B in Figure 1. In the figure, whenever two nodes are in the range of each other, there is a dotted 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 not in the range of any side of flow B, flows A and B are not contending any more. Therefore, the number of possible scenarios reduces to fifteen. Moreover, as known from 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 III-B and III-C). 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, neither 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.

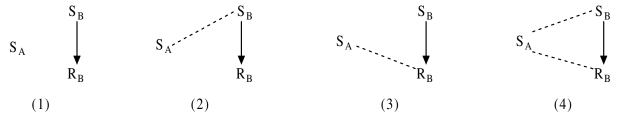

Fig. 1. Relationship between node S_A and Flow B

However, if the two flows just involve *three* nodes, the only scenario where the senders are not within range of each 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. We do not consider these scenarios since the fairness can simply be achieved by using fair queuing algorithms developed for the wire-line networks.

Now, we will show how the unfairness takes place in the above scenarios (figures 2, 3, and 4).

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 (the simulation results are presented in Section VII-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 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].

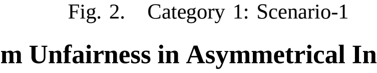

Fig. 2. Category 1: Scenario-1

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 contention than the other sender does. This has been referred as the asymmetrical information problem [12], which results in substantial long-term unfairness as explained below. Whenever S_A has completed transmission of a packet, both S_A and S_B begin to contend for the medium. In such a situation, the only

[1] It should be the sensing range if the sensing range is greater than the transmission range as in IEEE 802.11.

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

condition that S_A can win is that R_A responds with a CTS before S_B begins to send out a 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. 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 and thus 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.

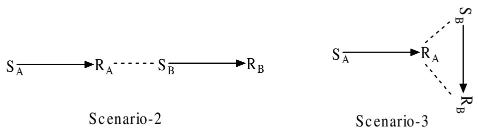

Fig. 3. Category 2: Asymmetrical Information Scenarios

Short-term Unfairness in Hidden-terminal Scenarios: In the hidden-terminal scenarios (Figure 4), once the RTS/CTS handshake 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 RTS/CTS 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 timer at the two hidden senders, and *Len* is equal to about 19 slots when the DSSS [11] is used. It is easy to see that the condition in (2) 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 VII-A). Consider the situation that the CWs at both the senders are very small (e.g., 31). As discussed above, under such situation, the transmission of RTSs is very likely to collide. The collision may occur several times until the CWs are large enough to allow either node (say, node S_A) to get control of the medium. Once S_A completes transmit a packet, it resets the CW and backs-off before initiating another transmission. However, the *freezing* back-off timer at node S_B may be large compared to the back-off timer at S_A, and thus S_A may transmit several more packets before S_B's back-off timer decrements to a small value. Clearly, the *freezing mechanism* in IEEE 802.11 leads to short-term unfairness.

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 again. 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 (i.e., several packet transmissions by node S_A, followed by collisions, and then transmissions by node S_A again) 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 incorporated in 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 contention on the medium.

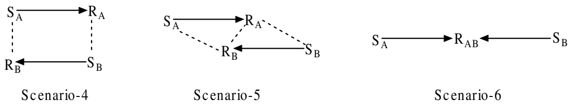

Fig. 4. Category 3: Hidden-terminal Scenarios

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 on the medium. In other words, the state information required for fair contention is *concealed* from the sender, and therefore we call it the *concealed information problem*. Here, the state information a sender needs is: when exactly the *contention period* occurs on the shared medium. A contention period *starts* after the transmission of an ACK frame on the medium, and *ends* when the transmission of a RTS frame begins on the medium.

In the scenario-1 (Figure 2), after one flow finishes the transmission of a packet (i.e., the receiver of the flow sends an ACK frame), the sender of the other flow can identify *neither* the beginning *nor* the ending of the contention period. In the hidden-terminal topologies (Figure 4), though the senders can identify the beginning of the contention period (i.e., whenever an ACK frame is transmitted on the medium, both the senders know this event), they cannot identify its ending (i.e., a sender does not know whether the other sender has already begun to transmit a RTS). In the asymmetrical information scenarios (Figure 3), whenever flow B transmits a packet on the medium, S_A can identify *neither* the beginning *nor* the ending of the contention period. Conversely, whenever flow A transmits a packet, both S_A and S_B cannot identify the ending of the contention period since they are hidden from each other.

B. 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. Our simulation results (presented in Section VII-D) show that flow B gets more bandwidth share than flow A. Let us consider the situation that nodes S_A and S_B choose the same back-off value before contending. Then, each of them initiates a RTS at the same time, resulting in a collision. Since node R_A is in the range of node S_B, R_A detects this collision and thus it will not send a CTS to S_A. On the contrary, since node R_B is out of the range of S_A, R_B will *not* detect the collision and therefore it will respond with a CTS to S_B. In summary, whenever there is a collision between the two RTSs, S_B *always* wins the contention, explaining the unfairness for flow A.

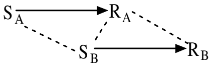

Fig. 5. Scenario-7: Imprecise Collision Detection Mechanism

C. Short-term Unfairness Due to High Contention

Now we 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[3] flows and all the six nodes are in the range of each other. To explain why there is short-term unfairness (the simulation results presented in Section VII-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 flows (say *n*) within the contention range as well as the CWs at the senders. A flow 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 their transmission. 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.

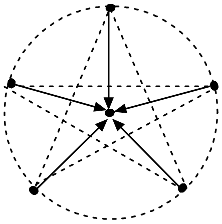

Fig. 6. Scenario-8: High Contention (all nodes hear each other)

D. Remarks on the MAC Fairness

So far, we have systematically identified the possible scenarios that show unfairness due to concealed information problem (Figures 2, 3, and 4), imprecise collision detection mechanism (Figure 5), or high-contention (Figure 6). Now we would make three important remarks on the fairness issue.

Remark 1: A *flow-contention graph* technique that has been popularly used in the literature is not an appropriate means for the study of MAC fairness. It is easy to see that all the scenarios with two contending flows discussed above will have the same flow-contention graph (see Figure 7), but totally different performance results are obtained for these scenarios. Therefore, any fairness scheme (e.g., [8], [17]) based on such a graph may not work very well.

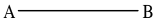

Fig. 7. Flow Contention Graph for Scenarios with two Contending Flows

Remark 2: In contrast to a purely sender-based protocol (e.g., IEEE 802.11), if we were to design a purely *receiver-based* algorithm, it would suffer from the similar problems because the two *receivers* may not be within the range of each other when two flows are contending with each other. Therefore, to achieve fairness, the protocol should employ some sort of cooperation between sender and receiver.

Remark 3: In a given scenario, whether short-term or long-term unfairness will occur largely depends upon the topology. When the topology involves asymmetry, long-term unfairness occurs. Conversely, in a symmetrical topology, the short-term unfairness occurs while the long-term fairness is ensured.

E. Impact of Unfairness

While the impact of long-term unfairness is obvious, as shown in [14] [22], the short-term unfairness at MAC layer may lead to a large delay jitter , and thus it substantially degrades the performance of the delay- or jitter-sensitive traffic (such as video and audio). The short-term unfairness also has a great impact on the adaptive traffic (e.g., TCP kind of traffic). Let us consider an example that there is a TCP flow from node S_A to S_B in the hidden-terminal scenario (see scenario-6 in Figure 4). From the simulation trace, we found that the TCP Data packets (from S_A to S_B) and the TCP Ack packets (from S_B to S_A) will always be transmitted in *bursts*. That is, in the steady-state, node S_A sends as many TCP Data packets as permitted by the congestion window. Then, node S_B transmits all the TCP Ack packets in a burst back to S_A. Through detailed analysis of the trace records, we found that this is mainly due to the short-term MAC unfairness between nodes S_A and S_B, i.e., once a node (say S_A) gets control of the medium, it may capture the medium for a long time before the other node (i.e., node S_B) transmits any packet. According to [14], the above phenomenon has also been observed in the wireless LAN (e.g., WaveLAN), and has been referred as *ACK compression*. The ACK compression results in packet loss at the congested nodes. Also, the link may be idle for some duration between the bursts of TCP Data and ACK packets.

In addition to the negative effects on jitter and TCP performance, the short-term unfairness may also 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, the MAC layer discards the packet

[3] Of course, one can choose a *larger* number of flows to show the short-term unfairness in a scenario with high contention.

and conveys this event to the routing protocol. As the routing protocol (e.g., AODV) interprets the discarding of a packet as an indication of link-breakage, the node discards all the packets in the queue and initiates a new route discovery process, though the link is still available. Even worse, in such a situation, the new 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 in the route discovery process itself. Therefore, a flow may be starved for a considerable time. In fact, the above phenomenon has been observed very frequently in our simulation when AODV is used. It is clear that it is extremely important to achieve both short-term and long-term fairness at the MAC layer.

IV. General Framework to Achieve MAC Fairness

In order to achieve MAC fairness in general, we propose a framework, which includes 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 may deviate from its fair share, a compensation model is needed to compensate the share of the user based on its past usage of the medium. *Thirdly*, we should have a distributed algorithm in place, which tries to realize the above two models, and thus achieves fairness.

A. Fairness Model

The objective of a fairness algorithm is to ensure that a *user* gets a fair share (say, ϕ_x for user x) over every certain duration (say T_{cyc}). Based on whether the *flows* or *nodes* are defined as *users*, *flow-level* or *node-level* fairness is desired. Since a node may have multiple flows associated with it, the flow-level fairness has finer granularity than the node-level fairness, and therefore we consider flow-level fairness here. 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. We aim to provide short-term fairness (implying long-term fairness). If the fairness is defined as the equal share among *n* active[4] flows (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 flows. Moreover, how to decide the fair share (ϕ_x for flow 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. Therefore, at the MAC layer, it is reasonable to assign ϕ_x with $1/n$ when *n* flows are active, implying that every *active* flow should transmit *exactly* one packet whenever *n* packets are transmitted over the medium. In summary, this can be formalized as,

$$\begin{cases} W_{cyc} = n \\ \phi_x = 1/n \end{cases} \tag{3}$$

W_{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., [7]), and thus are very *adaptive* to the *dynamic* network conditions.

B. Compensation Model

If the actual share (say, w_x for flow x) is always equal to the fair share ϕ_x during every window W_{cyc}, a system following the above fairness model behaves like a *dynamic* TDMA protocol, which has desirable fairness. By *dynamic*, we mean that the number of flows sharing the common medium is dynamic and that the transmission order among the flows from one cycle to another is also dynamic. However, in IEEE 802.11, w_x may deviate from ϕ_x, especially in the wireless ad-hoc networks due to the unfairness discussed in Section III. In particular, during a given window of length W_{cyc}, a flow may transmit more than one packet. We refer to this as the *over-use* of the medium by the flow. On the other hand, if a flow does not transmit any packet during the window, we call it *under-use*. Naturally, *normal-use* refers to the case when a flow 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 flow should adjust its rate as early as possible. Specifically, from the contention viewpoint, an active flow should be in one of the three modes: *aggressive*, *restrictive*, and *normal* at any given time. If a flow 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 flow has *over-used* the medium, it should enter the restrictive mode. However, if a flow gets its *fair* share, it should operate in the normal mode. Note that in the above compensation model we have not specified how much a flow should be compensated in the restrictive and aggressive modes, since that depends on the MAC protocol used as well as on the implementation.

Clearly, the above fairness and compensation models are based on the knowledge of *n* (the number of active flows) and w_x (the actual share of flow x). However, in ad-hoc networks, the sender as well as receiver of a flow has to *estimate* these two values in a *distributed* manner. Moreover, due to the distributed nature of ad-hoc networks, how to design a medium contention algorithm that can achieve fairness by utilizing the estimated information is a demanding challenge. The two issues are discussed in sections V and VI, respectively.

V. Estimation Algorithm

We first present a simple observation in IEEE 802.11. Whenever a flow transmits a *packet* over the medium, the *other*

[4] To recall, a flow is referred to be *active* whenever it has at least one packet waiting to send.

nodes within the contention range can generally overhear the RTS/Data, the CTS/ACK, or all the four *frames*[5]. For example, in the scenario of Figure 8, whenever flow A transmits a packet, node S_B can overhear the RTS/Data frames. In the hidden-terminal scenario (Figure 4), node S_B can overhear the CTS/ACK frames transmitted by flow A. 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 about overhearing is not always true. For example, in scenario-1 (Figure 2), whenever flow A transmits a packet, node S_B *cannot* overhear any of the four frames. This is also true for node S_A in the asymmetrical information scenarios (Figure 3) when flow B transmits a packet. For simplicity, we ignore these violation cases until to Section VI-B. Also, we assume that the sensing range is equal to the transmission range, and we will come back to this point in VIII-A.

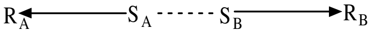

Fig. 8. Scenario where Senders Overhear RTS/Data

*A. Estimation of the number of active flows (*n*)*

In order to estimate the number of active flows within the contention range, every node maintains an ID list of the active flows. Whenever a node hears/overhears a frame (RTS/CTS/Data/ACK)[6], it inserts the corresponding *flow ID*[7] into the list if the ID does not exist in the list. In the case that the ID exists, the node simply refreshes the time of the entry containing this ID. With the above list, every node can easily get the number of active flows by counting the number of unique IDs in its list. Clearly, in such an algorithm, it is crucial to prevent stale entries, and therefore an entry is deleted after a timeout interval that is long enough to transmit a certain number (say W_e) of packets. 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 (say $n_e^{'}$) of *n*, therefore,

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

Clearly, the value of W_e affects the precision of n_e. Sometimes, n_e is greater than *n*, and we call this as an *over-estimation*. In contrast, the *under-estimation* refers to the case, where n_e is smaller than *n*. Both over-estimation and under-estimation may lead to bandwidth wastage. Specifically, when over-estimation occurs, the flows are restrictive and thus more slots are *idle*. On the other hand, when under-estimation occurs, the flows are aggressive and thus increasing the likelihood of *collision*. Now we consider three situations involving the imprecise estimation. *(i)* When a flow becomes active, the other nodes are *not* aware of this, until the flow transmits its *first* frame on the medium. This results in the under-estimation; *(ii)* After an active flow transmits its *last* packet in the queue and thus becomes inactive, its ID will *not* be deleted by the other nodes until a timeout interval long enough to transmit W_e number of packets. This results in the over-estimation; *(iii)* Due to short-term unfairness, an active flow may not be able to initiate any transmission in a given window W_e. As a result, though the flow is active, other nodes delete the corresponding ID, leading to under-estimation.

In our algorithm, whenever a flow becomes active, the flow enters the aggressive mode and thus transmits its first RTS immediately. Therefore, the under-estimation of the first case, if occurring, will last for a very short duration. In order to cope with the over-estimation problem in the second case, we should use a value of W_e as *small* as possible. On the other hand, a large value of W_e should be used to cope with the under-estimation problem in the third case. To cope with this dilemma, we propose *inactive-notification mechanism*. Specifically, whenever a flow sends out its *last* packet in its queue (i.e., switching from active to inactive), the sender of the flow uses a bit in the RTS/Data to tell other nodes that the flow is becoming inactive. Moreover, the receiver also piggybacks this notification in the *responding* CTS/ACK frames, as some nodes may not overhear the RTS/Data. All the nodes that hear the notification, rather than inserting or refreshing the flow ID, should *delete* this flow ID from the list. Clearly, 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, we found that the following values of W_e are better suited:

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

The above carefully designed timeout mechanism greatly prevents the imprecise estimation due to the following reasons: delayed transmission of the first frame of an active flow, stale entries, and short-term unfairness. However, it cannot solve the imprecise estimation due to the *absence* of overheard information in the violation cases (e.g., scenarios in figures 2 and 3). We will propose two novel algorithms in Section VI to cope with this problem.

B. Estimation of the actual share (w_x)

We now discuss how to estimate w_x, the actual proportion shared by flow x during the previous window W_{cyc}. This can be done by maintaining a transmission history at *each* node. Note that this transmission history is totally different from the ID list maintained for the estimation of n. We have intentionally designed the estimation algorithm in this manner so that it also applies when the sensing range is greater than the transmission range as discussed later in Section VIII-A. Now we discuss how to update the transmission history. Whenever a node hears/overhears a Data or ACK frame transmitted over the medium, the node will *append* the flow ID into its history. If a node overhears both the Data and ACK frames belonging to the same *handshaking* for a single packet, it should add

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

[6] Note that the correctness of the estimation does not rely on the overhearing of the Data/ACK frames.

[7] In fact, if we aim to achieve the node-level fairness for the senders as in IEEE 802.11 or if we assume that a node can act as the sender at the most for one flow, we can simply use the sender ID as the flow ID.

the ID only *once*. By maintaining such a history, a sender or receiver can easily know its flow's 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) $\{A, B, A, C, B, A, D, E, C, ...\}$ where A, B, C, D, and E are the flow IDs. Assuming that the n (estimated from a separate ID list) is equal to 5, then the estimated actual shares w_x for flows 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 (with *five* entries), i.e., $\{A, B, A, C, B\}$. In fact, in estimating w_x at the sender and receiver of flow x, the two nodes do not need to know the *exact* flow IDs of the packets transmitted by *other* flows. For example, when estimating w_A at S_A and R_A, the history $\{A, B, A, C, B\}$, if replaced by $\{A, -1, A, -1, -1\}$, will still work. Obviously, a node always knows the flow ID of a packet if the node is the sender or receiver of the flow. The above properties are very important when the Data/ACK is not interpretable to the overhearing nodes due to collisions or a large sensing range as discussed in Section VIII-A.

C. Detection of Unfairness

We now discuss how to detect the unfairness based on the deviation of a flow's actual share from its fair share. Let us again consider the example above (with transmission history $\{A, B, A, C, B, A, D, E, C, ...\}$). Since the fair share (ϕ_x) for each flow should be 1/5, flows A and B have over-used the medium during the latest window (with five entries), while flow C has had the normal-use of the medium. Flows D and E, however, have under-used the medium.

Based on the history, a sender or receiver of a flow can also measure *how much* the flow has over-used or under-used the medium recently. With this measurement, a flow 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 flow (through sender or receiver) checks its actual share in the latest transmission window (with n entries). 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 flow C in the above example). On the other hand, if the flow has under-used the medium in the window, N_a is initialized to *one*. In this case, the flow 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 is incremented by one. This process continues until the flow finds its share becoming normal in a transmission window. In the example given above, N_a at flows D and E are equal to 2 and 3, respectively. Similarly, we can get the value of N_r if a flow has over-used the medium during the latest transmission window. For instance, N_r at flows A and B are equal to 3 and 2, respectively.

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

Using the estimation algorithm proposed in the previous section, the sender as well as the receiver of a flow will obtain the following information: the number of active flows n, the actual share (w_x) of the flow, the contention mode that the flow should enter, and the corresponding degree of aggresiveness/restrictiveness. Clearly, the estimated values at the sender and the receiver of a flow may be different due to imprecise estimation. In this section, we will introduce a novel medium contention algorithm, called FMAC/CSR, which exploits the information available at the sender and receiver to achieve fairness. In FMAC/CSR, a node will act according to whether it is the sender or the receiver of the flow. Specifically, a sender contends for the medium with different priorities according to the mode and the corresponding degree that the flow should use. On the other hand, the receiver affects the sender's behavior in the contention through either an implicit or an explicit feedback.

In the design of FMAC/CSR, two issues need very careful consideration. *(i)* As mentioned, the decision (e.g., the mode that a flow should enter) made by the sender may be different from that made by the receiver. Therefore, rules must be defined that the sender and receiver should follow whenever such a discrepancy occurs. *(ii)* Since cooperation (between the sender and receiver of a flow, as well as among different flows) based on imprecise information is greatly involved in FMAC/CSR, the events such as medium being unduly idle, collisions, unnecessary explicit feedback, and deadlocks that degrade the throughput should be carefully avoided. This is the key to achieve fairness without unduly degrading throughput.

A. Sender-based Differentiated Access

To compensate for the under or over usage of a flow, a sender should contend for the medium with different priorities. In general, this can be realized using two different approaches. The first approach is that the sender will not transmit until the flow has the highest priority to access the medium among the contending flows. This is a *deterministic* approach, which achieves the desirable fairness, but potentially leads to substantial throughput degrade [12]. The second approach is to assign different contention windows to the senders. This is a *probabilistic* approach, which achieves a worse fairness but has a better throughput performance [7]. In order to achieve fairness without unduly degrading the throughput, we use a hybrid approach of the above two as will be clear below.

Now we define six rules that a sender should follow while contending for the medium whenever the medium becomes idle[8]. *(1)* If the flow should be in the aggressive mode, the sender will back-off by generating a random value from $[0, X]$ where $X = max(n, 2n - N_a)$. *(2)* If a flow should be in the normal mode, the random back-off value will be generated from $[2n, CW]$, where CW is the contention window. *(3)* If

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

the flow should be in the restrictive mode, the sender 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 sender will back-off by generating a random value from $[2n, CW \times N_r]$. *(4)* Irrespective of the mode that the flow is in, whenever the back-off timer expires and the medium is idle, the sender will transmit. *(5)* Whenever the medium becomes idle again after a busy state, the mode of the flow and the corresponding degree of aggresiveness/restrictiveness will be recomputed. Also, the back-off value will be regenerated. *(6)* The Contention Window (CW) is manipulated as in the Binary Exponential Back-off (BEB) of IEEE 802.11. The overall process followed at a sender is summarized in Figure 9.

Now we make three comments on the above rules. *(i)* Among the flows that are in the *same* mode but have different degrees of aggresiveness/restrictiveness, a *probabilistic* approach is adopted. For example, as known from the rule 1, a flow having a larger aggressive degree N_a will have a larger probability to transmit. Conversely, as known from the rule 3, a flow having a larger restrictive degree N_r will have a smaller probability in accessing the medium. *(ii)* Among the flows that are in the *different* modes, a *deterministic* approach has been used. For example, as seen from rules 1 and 2, the flows that are in the aggressive mode will get higher priority than those in the normal mode since X must be smaller than $2n$. However, our scheme is not *fully* deterministic since a flow, irrespective of the mode it is in, may transmit whenever the back-off timer expires as indicated by rule 4. This is crucial to prevent the throughput degradation when cooperation between flows is involved. Consider that there are two flows, and due to imprecise estimation, both of them operate in the restrictive mode. If a fully deterministic method is used, both the senders will defer since they are gracefully waiting for the other sender to transmit. Therefore, the medium will be unduly idle or even worse, a deadlock will occur, resulting in substantial throughput degradation. *(iii)* The rule 5 is completely different from the freezing mechanism used in IEEE 802.11, which is one of the main reasons of unfairness as discussed in Section III. On the other hand, as known from rule 6, the BEB algorithm is kept for its efficiency in resolving collision.

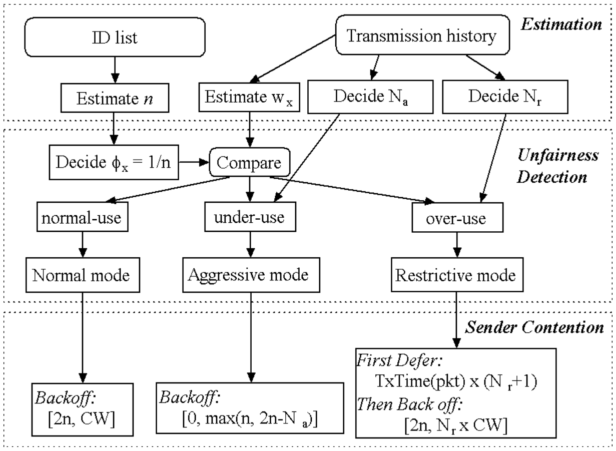

Fig. 9. Overall Process Followed at the Senders

B. Motivation to Use Receiver Feedback

The above sender-based mechanism can achieve fairness if the estimated information at a sender is always correct. However, as a sender of a flow may not overhear any frames transmitted by some other *contending* flows, two problems arise. The *first* one is that since the sender may get an incorrect estimation of the contention, it may not be aware of the situation whenever the flow has over-used/under-used the medium, and thus the compensation will not be performed. The *second* problem is that whenever one of the other contending flows transmits a packet over the medium, the given sender cannot identify the contention period, and thus it cannot contend efficiently. Due to the above two problems, a purely sender-based approach cannot always achieve fairness and sometimes it may even degrade the performance, as clearly shown by the following two examples.

In scenario-1 (Figure 2), since node S_A cannot overhear any of the frames transmitted by flow B, the n estimated at node S_A is always equal to one. Therefore, the S_A determines that the fair share for flow A should be one. Moreover, since the transmission history at node S_A will *never* contain flow B's ID, the actual share of flow A estimated at node S_A is also always equal to one. As a result, S_A will behave as if flow A was always in the *normal* mode. The above discussion also applies to S_B, and thus no compensation will be done. As a result, the sender-based algorithm in this topology will deliver the similar performance (e.g., substantial short-term unfairness) as that under IEEE 802.11. On the other hand, in the asymmetrical information topologies (see Figure 3), suppose that flow B has transmitted two or more packets consecutively, flow B will enter the restrictive mode and thus S_B will defer its transmission. However, since S_A cannot identify the contention period, it cannot contend efficiently. Specifically, when flow B is transmitting, S_A contends futilely, resulting in a large CW. However, after flow B finishes its transmission (i.e., a new contention period begins) and enters the restrictive mode, S_A will not take this opportunity to transmit since it is still deferring. Therefore, the medium will be *idle* for a *long* time and then flow B may get control of the medium *again*. Therefore, in this topology, the sender-based algorithm cannot achieve fairness and it even reduces the throughput of IEEE 802.11. All the above observations are confirmed by the simulation results in Section VII.

To solve the above problems, we recall that 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. Therefore, *in the case that the sender of a flow cannot overhear any frames transmitted by some other contending flows, the receiver must be able to overhear some frames transmitted by those contending flows.* In light of this, we propose two general *receiver feedback mechanisms*. In particular, whenever a receiver, based on its own understanding, notices that the flow has *over-used* the medium in the latest transmission window (and thus the flow should enter the restrictive mode), the receiver will try to slow down the sender by sending out a restrictive-notification to the sender. On the other hand, when a receiver finds that an *active*[9] flow has *under-used* the medium and now it is the right time (i.e., the beginning of a new contention period) for the sender to contend for the medium, the receiver may send out an aggressive-notification to expedite the sender's contention. However, when the receiver thinks that the flow has got a fair share, no special action is taken by the receiver. Clearly, in our FMAC/CSR, the sender still plays a more active role than the receiver does.

C. Receiver Feedback: Implicit Restrictive-notification

If a flow has over-used the medium, it must have just transmitted a packet. Therefore, the restrictive-notification can always be piggybacked in an ACK frame in an *implicit* manner. The remaining issues for this mechanism include: what information should be feedbacked, and what decision the sender should make after receiving the notification.

Regarding the issue of what the feedback information should be, there are several options: the number of active flows n, the actual share (w_x) of the flow, and the degree of restrictiveness N_r of the flow. Let us consider that the receiver feedbacks its estimated n to the sender. Therefore, a 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, in the scenario-1 (Figure 2), n used at node S_A is equal to two, and S_A will determine that the fair share for flow A should be 1/2, which reflects the real state of the contention. However, since the transmission history at S_A will *never* contain flow B's ID, S_A operates as if the flow was *always* in the *restrictive* mode. The same is true for S_B, leading to drastic throughput degrade. Therefore, the feedback of n may not be a good choice. We propose that the receiver should feedback to the sender with the degree of restrictiveness N_r, since N_r reflects both n and w_x. Of course, one can explore some other possibilities.

Whenever the sender receives a restrictive-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[10]. Otherwise, the sender will put itself in the restrictive mode with the degree contained in the notification.

D. Receiver Feedback: Explicit Aggressive-notification

If a flow has under-used the medium in the latest window, it must have *not* transmitted any packet recently, and there is no way to *piggyback* the aggressive-notification. Therefore, the aggressive-notification must be sent to the sender in an *explicit* manner. We can define a frame similar to RTS to carry the notification. Upon receiving the notification, if the sender is in the restrictive mode, it ignores the notification. Otherwise, it contends for the medium as per the aggressive mode with a degree of aggressiveness N_a that is contained in the notification.

Two issues need further investigation. *First*, we should try to avoid the collision of aggressive-notification frames as there may be more than one receiver intending to send out a notification at the same time. *Second*, since the *explicit* transmission of an aggressive-notification is a wastage of bandwidth, we should not use the aggressive-notification whenever it is unnecessary. In particular, if the sender is already aware that the flow should be aggressive and the sender can identify the contention period, the receiver should not send out a notification. Let us consider the scenario presented in Figure 10. The nodes R_A, R_B, S_C S_D, R_C, and R_D are within the range of each other. In such a scenario, the sender S_C have the same understanding of the contention as the receiver R_C does, and thus R_C should not send out a notification when flow C has under-used the medium. This is also true for R_D. In contrast, if flow A or B has under-used the medium, the notification should always be sent by R_A and R_B. Moreover, if there is another contending flow that is in normal/restrictive mode, the notification should be sent out *before* the sender of that flow transmits its RTS frame. In summary, for the *second* issue, we need to investigate some mechanisms, which can intelligently decide whether or not an aggressive-notification should be sent, and if it is to be sent, *when* to send it.

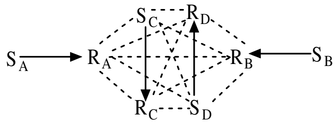

Fig. 10. Scenario Illustrating the Aggressive-notification

To cope with the first issue discussed above, before a receiver sends out an aggressive-notification, it will first back off by a random value. To cope with the second issue, the priorities for the medium access should be in the following order (from highest to lowest): senders of the flows in the aggressive mode, receivers of the flows in the aggressive mode, senders of the flows in the normal mode, and then, senders of the flows in the restrictive mode. Since the senders of the flows that are in aggressive mode will back-off with a maximum value equal to $2n$ as discussed in Section VI-A, the random back-off value for the aggressive-notification can be within the range $[2n, Y]$ where $Y = max(3n, 4n - N_a)$. Clearly, the maximum value a receiver will back-off before sending out the aggressive-notification is $4n$. Therefore, all the senders of the flows that are in normal/restrictive mode should generate a random back-off value which is at least $4n$ rather than $2n$ as used in Section VI-A. In this way, we avoid the unnecessary transmission of aggressive-notification but do not compromise on the effectiveness of this mechanism. For example, in the scenario of Figure 10, the receiver R_C will not send out an aggressive-notification whenever flow C should be in the aggressive mode since the sender S_C will send out a RTS before the receiver R_C's back-off timer expires. On the other hand, the receiver R_A will always send out a notification to its sender S_A whenever the flow A should be aggressive.

[9] Note that a receiver can know whether the flow is active with the help of the inactive-notification mechanism introduced in Section V-A.

[10] This is to prevent the unnecessary use of the receiver information in the scenario where the sender has complete information about the contending flows, e.g., in the high-contention scenario of Figure 6.

Summary of FMAC/CSR: In Table I, we summarize the FMAC/CSR. 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 restrictive-notification mechanism. Obviously, FMAC/CSR-3 refers to the FMAC/CSR-2 plus the aggressive-notification 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* Aggressive-notification
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:* Restrictive-notification	*Piggyback:* Restrictive-notification

VII. SIMULATION RESULTS

In this section, we present the simulation results to compare our FMAC/CSR algorithm with IEEE 802.11. The simulations were performed under the NS-2 with CMU wireless extensions [6]. All the assumptions described at the beginning of Section III apply in the simulation. The raw bandwidth is set to 2 Mbps and the packet of the CBR traffic is 1000-bytes long, leading to a *maximum* throughput of about 1.4 Mbps due to the overheads of IEEE 802.11. The sensing range is equal to the transmission range. The well-known Jain's index [5] is used as the main measure, which is defined as follows:

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

where *N* is the total number of flows that 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*. As the computation of γ_i depends on *w*, the value of Jain's index also depends on *w*, though *w* does not appear in the formula directly. Generally, 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 [14], 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 11. For 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 IEEE 802.11, FMAC/CSR-1 greatly improves the fairness. However, FMAC/CSR-2 and FMAC/CSR-3, which incorporates the receiver-feedback mechanism(s), do not show any advantage in comparison to FMAC/CSR-1. The reason is that the senders in this topology can get a precise estimation, and thus the receiver feedback mechanisms do not play any roles.

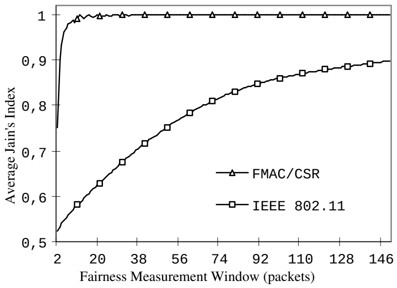

Fig. 11. Fairness Index in Hidden-terminal Topologies

Since *maximizing capacity utilization* and *achieving fairness* are generally two *conflicting* objectives in wireless ad-hoc networks [15], when evaluating a fairness algorithm, it is also necessary to look at the throughput results to ensure that the algorithm does not unduly degrade the throughput. Table II presents the throughputs under different schemes. While FMAC/CSR greatly improves the fairness, it also improves the aggregate throughput. 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 III-A, 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 FMAC/CSR, since the flows can cooperate in a distributed manner (e.g., when one flow 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 (Figure 3), we first present the throughput results in Table III. Clearly, under IEEE 802.11, flow B gets almost the entire bandwidth while flow A is almost starved. The reason of long-term unfairness has been explained in Section III-A. Under FMAC/CSR-1, the fairness substantially improves, however, it does not achieve long-term fairness completely. Moreover, the aggregate throughput degrades. The reason for these two observations has been explained in Section VI-B.

The fairness as well as the aggregate throughput substantially improves under FMAC/CSR-2, which incorporates the restrictive-notification mechanism. However, the performance does not improve any further under FMAC/CSR-3, which includes the aggressive-notification 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, flow B enters the restrictive mode and thus S_B 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 restrictive-notification mechanism is used (as in FMAC/CSR-2), node S_A will receive a notification and thus defer. On the contrary, without the restrictive-notification (as in FMAC/CSR-1), S_A will futilely retry. In other words, the restrictive-notification mechanism helps node S_A to identify the contention period, explaining the performance improvements in FMAC/CSR-2. In this manner, the restrictive-notification mechanism has accomplished the task (i.e., helping S_A to identify the contention period) that the aggressive-notification mechanism is supposed to do, and therefore the aggressive-notification mechanism introduced in FMAC/CSR-3 does not show any additional advantage.

The Jain's index is presented in Figure 12. 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

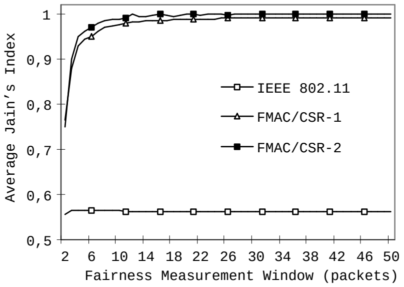

Fig. 12. Fairness Index in Asymmetrical Information Topologies

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 a 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 a 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 [10] can solve this problem, but two additional busy-tone channels are required. For simplicity, here we make a slight modification in 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 16 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 13. 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 restrictive-notification mechanism, greatly improves the fairness. However, under FMAC/CSR-2, the short-term unfairness remains and the aggregate throughput degrades substantially. The reason is again due to the fact that, when one flow enters the restrictive mode, the sender of the other flow cannot identify the contention period and thus cannot recognize the opportunity to transmit. The FMAC/CSR-3, which includes the aggressive-notification, improves the short-term fairness as well as the aggregate throughput compared to FMAC/CSR-2.

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

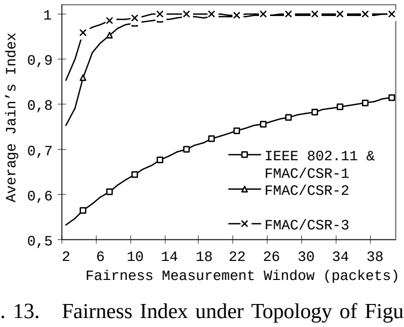

Fig. 13. Fairness Index under Topology of Figure 2

D. Scenario-7 of Figure 5

The throughput results are presented in Table V and the Jain's index is presented in Figure 14. 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 III-B. Under FMAC/CSR-1, the fairness substantially improves. Again, the FMAC/CSR-2 and FMAC/CSR-3, which incorporate the receiver-feedback mechanisms, do not show advantage in comparison to the FMAC/CSR-1 as the senders can get the complete information about the contending flows.

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

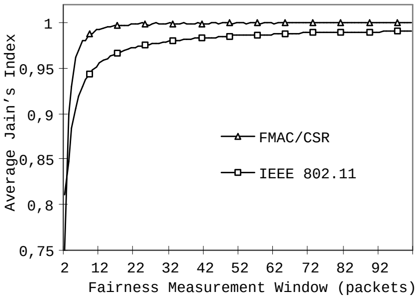

Fig. 14. Fairness Index under Scenario-7 (Figure 5)

E. High-contention topology

For the high-contention scenario shown in Figure 6, the Jain's index is displayed in Figure 15 and the throughput results are presented in Table VI. The IEEE 802.11 exhibits substantial short-term unfairness. In contrast, FMAC/CSR-1 greatly improves the short-term fairness compared to IEEE 802.11, and again the receiver feedback mechanisms do not play any role here.

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

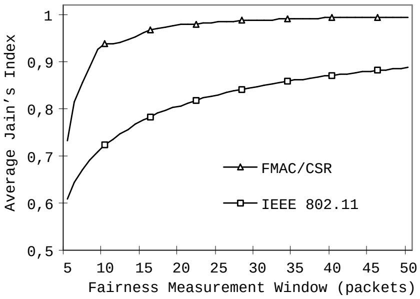

Fig. 15. Fairness Index in High-Contention Topology

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 16, which combines the main scenarios discussed so far. For convenience of explaining the relative positions of the nodes, we have drawn two concentric circles (dotted-line) 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 all the nodes on the inner circle but out of the ranges of all 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. The *Large-Col-EIFS* mechanism discussed before is adopted in this topology.

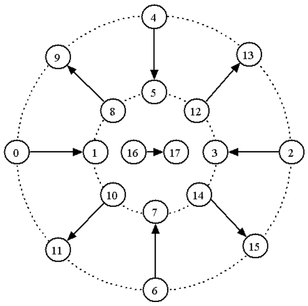

Fig. 16. Scenario-9: Complex Topology

The throughput results are presented in tables VII and VIII, while the Jain's index is displayed in Figure 17. 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 shown by the Jain's Index in Figure 17.

Since the above topology involves all the causes of unfairness discussed in Section III, the FMAC/CSR should be able to deliver similar performance (i.e., achieving fairness without unduly degrading throughput) in a randomly generated topology as long as all the single-hop flows are contending with each other.

TABLE VII

THROUGHPUTS IN COMPLEX TOPOLOGY (FIGURE 16)

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

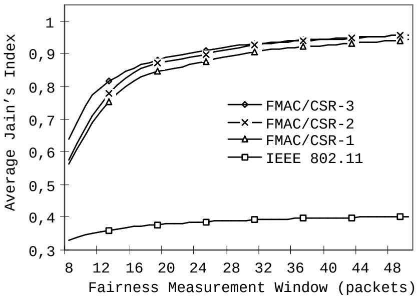

Fig. 17. Fairness Index in Complex Topology (Figure 16)

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* is quite precise except for the imprecise-estimation due to the absence of overheard information (e.g., in scenarios of figures 2 and 3). Here, we evaluate the precision of the estimate of *n* when the traffic is dynamically changing (i.e., exponential On/Off traffic) to show that the timeout mechanism designed in Section V-A can work well under the dynamic conditions. The topology of Figure 6 is used for study. 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. 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 18 presents the results during a typical duration of 2 seconds. We notice that n_e is equal to *n* most of the time.

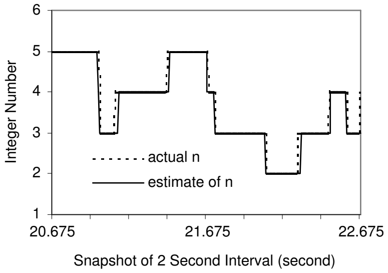

Fig. 18. Comparison between Actual and Estimation of *n*

VIII. DISCUSSION AND RELATED WORK

A. Further Comments on FMAC/CSR

B. Related work

The fairness problem in the random wireless MAC protocols was first highlighted in [2]. Recently, the fairness issue has been extensively addressed. Specifically, several works (e.g., [18] and the references therein) have addressed the unfairness problem due to the *location-dependent transmission error* in wireless LANs. More recently, assuming that there is no wireless error, a lot of works addressed the fairness issue in wireless ad-hoc networks. Based on the information needed and the fairness objective (global or local), these works can be categorized into two classes. The first category aims to design an ideal centralized scheduling algorithm, which 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., [15], [16]). Though these scheduling algorithms may achieve a network-wide fairness, they normally need global information (e.g., flow-contention graph or network topology). Moreover, whenever the network status changes (e.g., the topology changes due to mobility, or a flow enters or leaves the network), this information should be broadcasted in the network and a new scheduling decision has to be made. On the contrary, some other works (e.g., [7], [12], [17], [21]) aim to achieve fairness at the MAC layer itself. Our work belongs to this category. Though an algorithm following this approach may only achieve local fairness, it is desirable that it can operate in a distributed manner. Moreover, to *practically* implement an ideal centralized scheduling algorithm, a distributed MAC layer fair algorithm is required to approximate the ideal algorithm [15]. Therefore, whether the objective is to achieve local fairness or global fairness, MAC layer fairness is essential.

Now we review the works that followed the second approach. In [17], the authors have developed a fully distributed MAC protocol, called Proportionally Fair Contention Resolution, which dynamically adjusts the probability (based on its observation of the medium states, e.g., collision, idle, or busy) with which a sender accesses the medium. Based on the overheard information (rather than only the medium states), the works in [7] and [21] try to make the medium access more adaptive. Specifically, the work in [21] aims to emulate the Self-Clocked Fair Queueing [9] in wireless ad-hoc networks by adaptively choosing a back-off value proportional to the finish tag of a packet to be transmitted. The work in [7] dynamically tunes the contention window of a node based on its estimation of the sharing of the medium. All the above schemes are sender-based, which cannot *always* achieve fairness since the information at the sender is not always precise. On the other hand, the authors in [12] have exploited the receiver information to implement a FIFO queue among the contending nodes. However, they have considered the asymmetrical information topology only, and how the sender and receiver should cooperate is largely unaddressed. Moreover, their scheme results in substantial throughput degrade as a fully deterministic method is adopted. Even worse, as shown in [20], due to the deterministic nature, deadlocks are likely to occur in a complex scenario. In the preparation of this paper, we are aware of another work [8] that aims to achieve MAC fairness by generalizing the approach in [17] using a game theory framework. However, as in [17], it still follows a purely sender-based approach, and it derives the fairness algorithm based on a flow-contention graph, which is inappropriate as remarked in Section III-D. While all the above works (except [17]) rely on the correctness of the estimation based on the overheard frames, but none of them describes how to estimate correctly when the sensing range is greater than the transmission range in which case the overheard frames may not always be interpretable and thus the estimation is not precise.

Compared to all the above works on MAC fairness, our FMAC/CSR is quite unique due to the following reasons: (1) it exploits the information available both at the sender and the receiver, (2) it achieves fairness without degrading the throughput in the scenarios where spatial reuse is not possible, and (3) it is applicable even when the sensing range is greater than the transmission range.

IX. Conclusions

In this paper, we have proposed a novel medium access control protocol, called FMAC/CSR, which exploits information available at the senders and receivers to achieves MAC fairness. The simulation results show that the FMAC/CSR greatly improves the fairness without unduly degrading the throughput.

The main contributions include: *(i)* identification of general 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* flows as well as the actual share of a flow; *(iv)* proposal of the FMAC/CSR, which incorporates a suite of mechanisms: differentiated access at the senders, restrictive-notification, and aggressive-notification.

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] 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 Backoff Algorithm for IEEE 802.11 DFWMAC,” ACM MOBIHOC, 2002.

[8] Z. Fang, B. Bensaou, “Fair Bandwidth Sharing Algorithms based on Game Theory Framework for Wireless Ad-hoc Netoworks,” in IEEE Infocom, 2004.

[9] S. Golestani, “A Self-clocked Fair Queueing Scheme for Broadband Apllicaiotns,” IEEE Infocom, 1994.

[10] 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

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

[12] 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.

[13] 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.

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

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

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

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

[18] T. Nandagopal, S. Lu, V. Bharghavan, “A Unified Architecture for the Design and Evaluation of Wireless Fair Queueing Algorithms,” ACM MOBICOM, 1999.

[19] 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

[20] 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/

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

[22] 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