Achieving MAC Fairness in Wireless Ad-hoc Networks using Adaptive Transmission Control

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

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

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

Abstract*— Achieving MAC layer fairness in multi-hop wireless ad hoc networks is a very challenging issue. In this paper, we first show that IEEE 802.11 exhibits substantial short-term unfairness due to the hidden-terminal problem, high contention, and the concealed information problem (introduced in this paper). To achieve the short-term fairness, we propose Adaptive Transmission Control (ATC) algorithm as follows. Each node, in a distributed manner, estimates the *number of active nodes* within the contention range, as well as its *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 short-term unfairness, the actual share of a node may deviate from its fair share. To compensate for the bandwidth usage in the recent past, the node enters one of the three modes, *aggressive*, *restrictive*, and *normal*, indicating how the node should behave when contending for the medium. The contention window (CW) of the node is tuned according to the mode it enters. Due to the freezing mechanism of the IEEE 802.11, the tuning CW mechanism has no effect on the nodes that do not generate a new back-off timer when they contend for the medium. Therefore, we propose early-reset mechanism, which stops the back-off timer and resets the CW whenever a node enters the aggressive mode while containing a non-zero back-off timer. Due to the concealed information problem, the tuning CW and early-reset mechanisms still cannot achieve fairness in certain scenarios. Therefore, we propose receiver-coordination mechanism. Extensive simulation results show that the proposed ATC substantially improve the short-term fairness without unduly degrading the throughput.

I. INTRODUCTION

Recently, wireless ad hoc networks have attracted considerable research interest. As IEEE 802.11 [8] 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. As DCF operates in a *distributed* manner without having precise information about 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 [11]. In particular, in certain scenarios, though the bandwidth allocation is fair in a long-term, it is very unfair if we view the system from a short-term viewpoint. The short-term fairness is very important for the adaptive traffic (e.g., TCP kind of traffic) and for the delay- or jitter-sensitive traffic [11]. In this paper, we mainly focus on the topologies in which the short-term unfairness occurs but the long-term fairness is ensured.

Fairness can be achieved through two different ways. One is to design a scheduler (e.g., [12]), 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., [14]). We here follow the second method. We first show that the IEEE 802.11 exhibits substantial short-term unfairness due to the hidden-terminal problem, high contention, and the concealed information problem (explained in this paper later). To achieve fairness, we then propose the Adaptive Transmission Control (ATC) algorithm. In the ATC, every node, in a *distributed* manner, estimates the number of *active* stations (say *n*) within the contention range, as well as the actual bandwidth (say w_i) 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_i). Due to the short-term unfairness, the actual share (i.e., w_i) may deviate from the fair share (i.e., ϕ_i). To make the w_i as close to ϕ_i 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. The contention window (CW) of the node is tuned according to the mode it enters and the amount by which w_i differs from ϕ_i. However, due to the freezing mechanism used in the IEEE 802.11, the tuning CW mechanism does not affect the nodes that do not generate a new back-off timer when they contend for the medium. Therefore, we propose early-reset mechanism, which stops the back-off timer and resets the CW whenever a node enters the aggressive mode while containing a non-zero back-off timer. Due to the concealed information problem in the *multi-hop* networks, the tuning CW and early-reset mechanisms still cannot achieve fairness in certain scenarios. Therefore, we propose receiver-coordination mechanism, which exploits the information available only at the receivers, and helps the senders make more rational decisions. Extensive simulation results show that the proposed ATC substantially improve the short-term 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 short-term unfairness. We then, in Section III, present the adaptive transmission control (ATC) algorithm. The simulation results are presented in Section IV. The related work is reviewed in Section V, and the paper is concluded in Section VI.

II. Short-term Unfairness in IEEE 802.11

A. Basic Techniques in IEEE 802.11 DCF

The DCF defines two methods in accessing the medium: the two-way handshake and the four-way handshake. In the two-way handshake, the sender transmits a Data frame to the receiver, which responses 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.

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.

$$BackoffTime = Random() \times SlotTime \tag{1}$$

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 transmitting the packet and thus the medium becomes idle, all the 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. Contrary to a successful transmission, when a collision occurs, all the colliding nodes will generate a new random value.

B. Short-term Unfairness Due to Hidden-terminal

Figure 1 depicts the hidden-terminal problem. Since nodes A and C are hidden from each other, they may simultaneously try to communicate with the node B, resulting in a collision. If the two-way handshake is used, the throughput will degrade drastically due to the collision of the Data frames, which are normally very long. On the contrary, if the four-way handshake is used, once the RTS/CTS has been completed successfully, the hidden-terminal problem does not arise any more. For example, in Figure 1, once node B sends back a CTS to node A, node C 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 nodes may collide, unless the following condition is satisfied,

$$| Z | > Len = TxTime(RTS) + SIFS \tag{2}$$

where *Z* is the difference between the back-off timers at the two hidden nodes, 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 [8] 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 presented in Section IV.A). Consider the situation that the CWs at nodes A and C 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 nodes A and C may collide. The collision may occur several times until the CWs are large enough to allow either node to get control of the medium. In particular, one of the two nodes (say, node A) may select a small back-off time from its CW, while the other node (i.e., C) selects a large value. The difference between the two values may be large enough to satisfy the condition, and thus node A can successfully transmit a packet. Once the packet transmission is completed, node A resets its CW and backs-off before initiating another transmission. However, the remaining back-off timer at node C may be large compared to the back-off timer at node A, which is drawn from the range [0, CW_{min}]. In that case, node A may transmit several more packets before node C's back-off timer decrements to a small value.

Whenever the back-off timer at node C becomes small, C contends for the medium. However, as the CW at A is equal to CW_{min}, the contention is most likely to result in a collision. After the collision, node A doubles its CW from CW_{min} whereas C doubles its CW from a larger value (at least 63). Therefore, node A is more likely to get control of the medium *again*. This is obviously unfair for C. Moreover, this process may repeat several times, leading to starvation at node C for a long period (compared to the time needed for a packet transmission).

However, several mechanisms incorporated in the IEEE 802.11 prevent node C from starving *completely*, such as: *(i)* after every packet transmission, node A will back-off before initiating another transmission, which gives node C a chance to contend for the medium; *(ii)* the CW at node C will be reset to CW_{min} after the retry limit is reached. These mechanisms

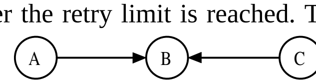

Fig. 1. Hidden-terminal Scenario

ensure *long-term* fairness between the two flows.

C. Short-term Unfairness Due to High Contention

The short-term unfairness also occurs in a scenario where all the nodes are within the range of each other but the contention is very high. For example, in the scenario shown in Figure 2, 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 IV.B), 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 all 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. In summary, though the BEB is effective in resolving collisions, it results in shot-term unfairness when the contention is high.

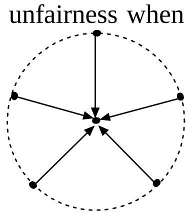

Fig. 2. High Contention Scenario

D. Short-term Unfairness Due to Concealed Information

Another representative scenario exhibiting short-term unfairness (the simulation results presented in Section IV.C) is shown in Figure 3, which has also been discussed in [1]. The *dashed* line between nodes B and C indicates that the two nodes are in the transmission range of each other. Consider the situation that node A is in control of the medium and thus node C is not able to respond to any request from node D. However, since node D does not know about the ongoing transmission between nodes A and B, node D may futilely retry, resulting in a large CW at the node. As for the node A, after it transmits the packet, it will reset its CW and contend for the medium again. Since the CW at node D becomes very large, node A may transmit several packets *consecutively* before node D gets control of the medium, leading to short-term unfairness. However, the mechanisms incorporated in IEEE 802.11, which have been described at the end of Section II.B, prevent node D from starving *completely*, and also ensure long-term fairness between the two flows[1]. As pointed out in [1], the key problem in such scenarios is that the senders (e.g., node D) cannot identify the *contention period* (i.e., the duration after the ACK frame and before the next RTS frame on the medium). In other words, the state information of the *medium* is *concealed* from the sender, and therefore we call it the *concealed information problem*. Here the information, that is needed, is the time when the contention period starts and ends. Note that the *asymmetrical information problem* introduced in [10] is indeed a special case of the concealed information problem introduced here. However, in the asymmetrical information problem scenario, one sender suffers from the concealed information problem, while the other sender does not, and thus resulting in *long-term unfairness* between the two flows.

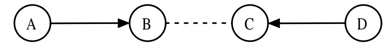

Fig. 3. Concealed Information Topology

In the hidden-terminal topology (see Figure 1), though the senders can identify the contention period, substantial short-term unfairness is there. This is because the transmission of the RTS by the hidden terminal (say node C) is *concealed* from the sender (i.e., node A). Therefore, the hidden-terminal problem is also a special case of the concealed information problem, where the *definition* of the concealed information includes the information of the transmission of the RTS.

III. ADAPTIVE TRANSMISSION CONTROL (ATC)

In this section, we propose an adaptive transmission control (ATC) algorithm, which aims to achieve MAC layer fairness in the multi-hop wireless ad hoc networks.

A. Objective of ATC

The objective of a fairness algorithm is to ensure that a node gets a fair share (say, ϕ_i for node *i*) during every certain long 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 can achieve the long-term fairness, which however does not imply short-term fairness. On the other hand, if T_{cyc} is small, the algorithm can 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* in this scenario. Obviously, T_{cyc} or W_{cyc} is more complex to define if the share is not equal for all the active nodes. Moreover, how to decide the value of the share (ϕ_i for node *i*) is in itself an issue. At a *higher* layer, the value of ϕ_i should be determined based on the application requirement. However, as our focus is to achieve fairness in the contention for the shared medium, we

[1]According to the results presented in [1], in such a scenario, node D will be completely starved in the MACAW protocol.

simply assume that all the applications have the same weight. Therefore, at the MAC layer, it is reasonable to assign ϕ_i 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 formulized as,

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

T_{cyc} and ϕ_i, 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.

If the actual share (say, w_i for node *i*) is always equal to ϕ_i during every window W_{cyc}, the system behaves like a dynamic TDMA protocol, which has the *ideal* fairness. However, the w_i may deviate from ϕ_i, especially in the wireless ad hoc networks due to the issues 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. The *objective* of our ATC, is to adjust the rate of all the active nodes as early as possible to compensate for the over-use and under-use in the previous window, and thus make w_i as close to ϕ_i as possible to achieve short-term fairness.

In order to achieve the above objective, 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 ATC is based on the knowledge of *n* (the number of active nodes) and w_i (the actual share of node *i*). However, in the ad hoc networks, every node has to dynamically estimate these two values in a *distributed* manner, which is discussed in the next subsection.

B. Estimation Algorithm

Whenever a node transmits a *packet* over the medium, other nodes within the contention range can generally overhear the RTS/Data, the CTS/ACK, or all the *frames* (note that the word '*packet*' implies the protocol data unit (PDU) of a higher layer whereas '*frame*' is the MAC layer PDU). For example, in the scenario shown in Figure 4, whenever node B transmits a packet to node A, node C can overhear the RTS/Data frame. In the hidden-terminal scenario (see Figure 1), whenever node A transmits a packet to node B, node C can overhear the CTS/ACK frame. In the scenario that all the nodes are within one-hop (e.g., scenario in Figure 2), every node can overhear *all* frames. However, the above observation is not always true. For example, in the concealed information scenario (see Figure 3), whenever node A transmits a packet to node B, node D *cannot* overhear any of the four frames. In order for the simplicity of the presentation, we will exclude this violation case, until Section III.E.

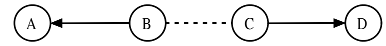

Fig. 4. Four Nodes with Two Flows Scenario

*1) 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, which is set as follows,

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

where $n_e^{'}$ is the previous estimate of *n*. With the above list, every node can easily get the number of active nodes by counting the number of IDs in its list.

2) Estimation of the actual share (w_i) : We now discuss how to estimate w_i, the actual proportion shared by node *i* during the previous window W_{cyc}. This can be done by maintaining a transmission history[2] 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_i during the latest window by simply checking how many times (say *m*) its *own* ID appears in the window W_{cyc}. Therefore,

$$Estimation(w_i) = m/n \tag{5}$$

As an example let us consider the history (beginning with the most recent entry) $\{1, 1, 2, 3, 1, 3, 2, 1, ...\}$ where 1, 2, and 3 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_i for nodes 1, 2, and 3 are 2/3, 1/3, and 0, respectively, since we need to look at first three entries in the history, i.e, $\{1, 1, 2\}$. In fact, in estimating w_i at node i, 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_1, node 1 does not need to know the IDs of the sender of the packets transmitted by nodes 2 and 3. 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 the Data/ACK is not interpretable due to collisions or large sensing range as discussed below.

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

3) Estimation when Sensing Range > Transmission Range: So far, for the convenience of presentation, we have simply assumed that the transmission range (TR) is equal to the sensing range (SR) as done in [6] and [10]. In practice, the SR may be greater than the TR. Therefore, our estimation algorithm, as well as other algorithms (e.g, [6], [10], and the Virtual Carrier Sensing mechanism in IEEE 802.11) based on overhearing may not work.

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. 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 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. With this observation in mind, we only need to make a slight modification in the estimation of actual share w_i when SR>TR. 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[3]. To cope with the situation that a node can detect both the Data and ACK frames belonging to a given 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 overheard. As we have mentioned before, the estimation of w_i at node i 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. Note that we need not make any modification in the estimation of *n* since it only relies on the overhearing of RTS/CTS.

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 (presented in Section III.E), 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, which is not very likely. Alternatively, we can estimate the *n* using some heuristics method as done in [3] and [4].

[3]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.

C. Tuning CW based on the Estimation

As discussed in Section III.A, based on the estimation of *n* and w_i, a node enters one of the following modes: aggressive, restrictive, and normal to compensate for the over or under usage in the immediate past. The compensation can be realized using two different approaches. One is to make the node that is in the restrictive mode, to defer its transmission until it enters the normal mode. This is a *deterministic* approach, which achieves the desirable fairness, but leads to substantial capacity degrade as shown in [10]. The other approach is to assign larger contention windows (CWs) to the nodes that are in the restrictive mode. This is a *probabilistic* approach as a node with a larger CW may still generate a small back-off timer. We follow the *probabilistic* approach.

In the IEEE 802.11, whenever the medium becomes idle (after a collision or after a successful transmission of a packet), a node may contend for the medium by generating a new back-off timer, or directly using the *frozen* back-off timer that has been generated previously. We first discuss the nodes that generate a new back-off timer. This occurs for two kinds of nodes: the nodes who have just experienced a collision, and the node who has just transmitted a *packet* successfully. When a node generates the back-off timer, rather than using the CW *directly*, we propose that the node should use a scaled value $S \times CW$, where *S* is the scaling factor determined by the node's mode. The *S* can be simply set to unity when the node is in the normal mode. When the node is in the aggressive mode, *S* should be smaller than one. On the other hand, when a node is in the restrictive mode, the *S* should be greater than one. Therefore,

$$S = \begin{cases} 1 & normal\ mode \\ 2 & restrictive\ mode \\ 1/2 & aggressive\ mode \end{cases} \tag{6}$$

However, the *S*, if defined as above, cannot differentiate between the nodes that have different *degrees* of over-use or under-use. Therefore, a node also records the number of times that it has been in the aggressive or in the restrictive modes since the *latest* occurrence of the normal mode, which are represented by $N_{aggressive}$ and $N_{restrictive}$, respectively. Since a node cannot enter aggressive mode from the restrictive mode (and vice-versa) without passing through the normal mode, $N_{aggressive}$ and $N_{restrictive}$ are reset to zero whenever the node's mode becomes normal. If a node is in the aggressive mode, the $N_{aggressive}$ is incremented by one when any *other* node transmits a packet. On the other hand, if a node is in the restrictive mode, the $N_{restrictive}$ is incremented by one when the node *itself* transmits a packet. Therefore, the scaling factor *S* should be modified as,

$$S = \begin{cases} 1 & normal\ mode \\ 2 \times N_{restrictive} & restrictive\ mode \\ 1/(2 \times N_{aggressive}) & aggressive\ mode \end{cases} \tag{7}$$

To limit the scaling factor, whenever $N_{aggressive}$ (or $N_{restrictive}$) reaches a maximum value (e.g., 8 in our implementation), it is not allowed to increase any further.

Several points should be noted in the above discussion. The *first* one is that $N_{aggressive}$ and $N_{restrictive}$ will be updated *only* after a successful transmission over the medium, and whenever, there is a collision, they remain unchanged. The *second* point is that in our algorithm, we still keep the main feature of the Binary Exponential Back-off (BEB) for its efficiency in resolving collision. For example, as in the BEB, after a collision, the CWs at the colliding nodes will be doubled, while the CW is reset after a successful transmission. The *third* point is that we adopt the scaled CW (i.e., $S \times CW$) to achieve fairness, rather than *directly* manipulating the CW. The reason is that the CW gives a good indication of the contention degree; otherwise we will lose the measure of how intense the contention is. For example, consider that a node enters the normal mode from restrictive, and then it experiences a collision. If we increase the actual value of the CW when the node is in the restrictive node, then after the collision, the CW will be doubled from a value that is *larger* than what should be to reflect the contention degree. In summary, since a *separated* scaling factor is used to achieve fairness, the ATC can also resolve collisions in an effective manner.

D. Early Reset Mechanism

Whatever has been discussed only applies to the nodes that generate a new back-off timer while contending for the medium. However, if a node already has a non-zero back-off timer, the scaling factor *S* does not have any effect since the node will *not* generate a new back-off timer from the $S \times CW$. This will lead to a problem if the node is supposed to enter the *aggressive* mode. Consider that a node generates a large back-off timer, and waits for the back-off timer to expire. During the waiting period, due to the freezing mechanism, other nodes may transmit more than *(n-1)* packets, and therefore, the node should enter the aggressive mode. However, the back-off timer may still be greater than zero and therefore it has to wait further, which potentially keeps the medium *idle* as other nodes are in the restrictive mode.

To solve the above problem, we introduce *early reset mechanism*. In particular, whenever a node is supposed to enter the aggressive mode from the *normal* but it still has a non-zero back-off timer, it should stop the current back-off timer, reset the contention window (CW), and then generate a new back-off timer from the $S \times CW$ window. Note that the early reset mechanism is not used if the node is already in the aggressive mode.

The adaptive transmission control (ATC) algorithm described so far can be summarized by Figure 5.

E. Receiver Coordination

Now we address the problem for the violation case described in Section III.B, where a node can hear *neither* the Data *nor* the ACK frame. For example, in the topology

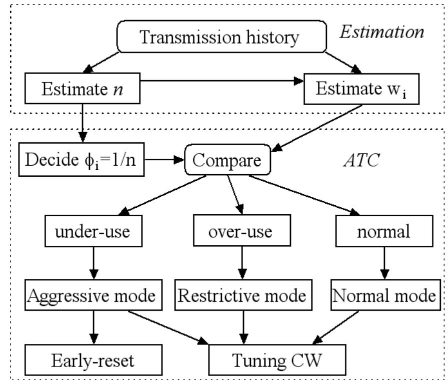

Fig. 5. Overall Process of ATC at the Sender

presented in Figure 3, since node A cannot overhear any of the frames exchanged between nodes D and C, n_e estimated at node A is always equal to one. Similarly, n_e at node D is also always equal to one. Therefore, both nodes A and D will *always* operate in the *normal* mode, and thus, in this topology, the algorithm proposed thus far will show the *same* performance as that under the IEEE 802.11.

In order to cope with the above problem, we notice that the receiver node C can overhear the ACK frame whenever node A transmits a packet to node B. Therefore, n_e estimated at node C is equal to two. This is also true for the node B. In light of this, 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 previous window, the receiver should 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, 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 A and D (see Figure 3) is equal to two, which reflects the real state of the medium. However, since the transmission history at node A will *never* contain node D's ID, node A will *always* operate in the *restrictive* mode. The same is true for D, 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 [10] in a slightly different context, and we choose the piggyback method to regulate the sender's rate. Whenever the sender receives the notification from the receiver, in addition to the back-off process as defined in the IEEE 802.11, the sender defers for a certain long time (say T_{defer}). The T_{defer} in our implementation is set to *TxTime(packet)*, which is the time needed to send out a *packet* including overheads of RTS, CTS and ACK. Note that the sender should ignore the notification if it is already in the restrictive mode.

IV. Simulation Results

In this section, we present the simulation results to compare the proposed adaptive transmission control (ATC) scheme with the IEEE 802.11. The simulations were performed under the NS-2 with CMU wireless extensions [5]. 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 [11] is used as the main measure, which is defined as follows:

$$F_J = (\sum_{i=1}^{N} \gamma_i)^2 / (N \sum_{i=1}^{N} \gamma_i^2) \quad (8)$$

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 [11], the index has been *averaged* over all *sliding* windows of *w* packets, which occur in the simulation run.

To study the improvements contributed by various mechanisms proposed in Section III, we divide the ATC into three different categories where each has different *extent* of the mechanisms as shown in Table I.

TABLE I

THREE CATEGORIES OF ATC

Scheme	Mechanisms
ATC-1	Estimation, Tuning CW
ATC-2	ATC-1 plus Early Reset
ATC-3	ATC-2 plus Receiver Coordination

A. Hidden-terminal topology

The Jain's index for the hidden-terminal topology (of Figure 1) is presented in Figure 6. 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 ATC-1 greatly improves the fairness as shown in the figure. The ATC-2, which incorporates the early reset mechanism, improves the fairness *further*. However, in this topology, the ATC-3, which further incorporates the receiver coordination mechanism, does not show any advantage in comparison to the ATC-2. The reason is that the senders in this topology can get the complete information about the contending flows, and thus the receiver coordination mechanism does not play any role.

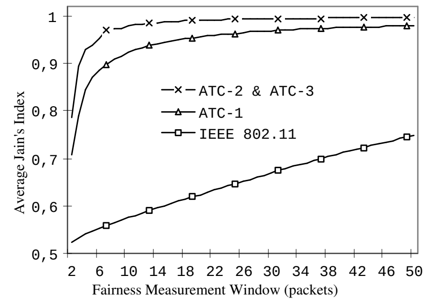

Fig. 6. Fairness Index in Hidden-terminal Scenario

The *long-term* fairness can also be reflected by the *average* throughputs, which have been tabulated in Table II. We notice that, under each of the schemes, the average throughput of the two flows are quite close to each other, which implies that the long-term fairness is achieved in all the schemes. However, the *standard deviation* of the throughput under different schemes is quite different, implying that different magnitude of *short-term* fairness has been achieved. In fact, the fairness conveyed by the *average* and the *standard deviation* has already been reflected by the Jain's index F_J with varying *w*.

While the ATC greatly improves the fairness, from the aggregate throughputs presented in Table II, we see that, the throughput degrades by about 10%. This shows the fundamental *conflict* between *maximizing capacity* and *achieving fairness* (see [12]). In particular, under IEEE 802.11, as explained in Section II.B, whenever a node (say node A) gets control of the medium, it can transmit packets consecutively without experiencing any collision. On the contrary, under the ATC, since the other node (i.e., node C) is in the aggressive mode, it will contend with node A, which may lead to collisions since the condition presented in equation (2) is difficult to achieve. In fact, in the hidden-terminal topology, to improve the fairness, the node that has under-used in the recent past has to contend aggressively even at the cost of the collisions. However, note that the throughput degradation under ATC is much less than that under the scheme presented in [10], which is typically more than 25%. This shows the advantage of using the *probabilistic* approach as discussed in Section III.C. Also note that the aggregate throughput under ATC-2 is slightly better than that under the ATC-1, which is attributed to the early reset mechanism.

TABLE II

THROUGHPUT UNDER HIDDEN-TERMINAL TOPOLOGY

Throughput (M bps)		IEEE 802.11	ATC-1	ATC-2	ATC-3
Average	A to B	0.678	0.609	0.615	0.615
	C to B	0.676	0.604	0.613	0.613
Standard deviation	A to B	0.239	0.057	0.022	0.022
	C to B	0.238	0.058	0.022	0.022
Aggregate		1.354	1.213	1.228	1.228

B. High-contention fully-connected topology

In this subsection, we present the results for the high-contention scenario shown in Figure 2. The Jain's index is displayed in Figure 7. Again, the IEEE 802.11 exhibits substantial short-term unfairness. On the contrary, the ATC-1 greatly improves the short-term fairness compared to the IEEE 802.11, and the ATC-2 improves it *further*. The receiver coordination mechanism again does not play any role in this topology.

Table III shows that the aggregate throughput degradation under ATC-1 is negligible compared to the IEEE 802.11. The ATC-2 even slightly increases the aggregate throughput, which is again an attribute of the early reset mechanism that facilitates *distributed* cooperation.

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

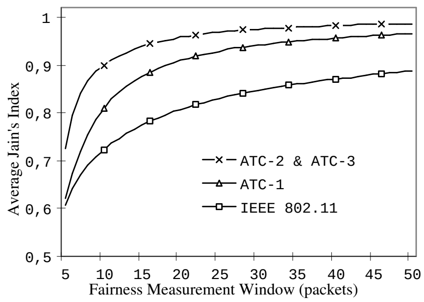

Fig. 7. Fairness Index in High-Contention Topology

TABLE III

THROUGHPUT IN HIGH-CONTENTION TOPOLOGY

Throughput (Mbps)	IEEE 802.11	ATC-1	ATC-2	ATC-3
Aggregate	1.410	1.408	1.416	1.416

C. Four-nodes topology

In this subsection, we present the results for the two topologies shown in Figure 3 and Figure 4.

Concealed Information Topology of Figure 3: 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 A sends out a RTS to node B, and then node B sends back the CTS. As node D is unaware of this CTS, it may send out a RTS, leading to a collision at node C. A node (e.g., node C) 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 the node A, it will transmit the Data frame. During the transmission, D may initiate its RTS again. If the RTS arrives at node C after the EIFS deferment is over, C will respond with CTS, which will collide with the Data frame from node A to B, and thus node B will discard the Data frame and defer by an EIFS. Now, for node D, since the RTS/CTS is successful, it transmits the Data frame. The Data frame is likely to be destroyed by node B's CTS in a similar manner, resulting in substantial bandwidth wastage. The dual busy tone multiple access (DBTMA) proposed in [7] can solve this problem, but two additional busy-tone channels are required. However, 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 EIFS duration, it will defer for a large enough duration enabling transmission of the Data frame. We call this as the *Large-Col-EIFS* mechanism. In this subsection, we present the results assuming that this mechanism is included in all the schemes.

The Jain's index is presented in Figure 8 and the aggregate throughput in Table IV. We notice that the aggregate throughput in IEEE 802.11 with the *Large-Col-EIFS* mechanism greatly improves. As for the fairness, IEEE 802.11 exhibits substantial short-term unfairness. Moreover, the ATC-1 and ATC-2 do not improve the fairness, which has been discussed in Section III.E. On the contrary, the ATC-3, which incorporates the receiver coordination mechanism, greatly improves the fairness. However, under the ATC-3, the fairness index still does not approach to unity even when the w is large (e.g., 50), implying that the unfairness remains. Moreover, the aggregate throughput degrades by about 25%. The reason is as follows. Consider that the receiver (say node B) notices that the sender (i.e., node A) has been aggressive in the recent past, and thus B piggybacks an over-use notification. Upon receiving the notification, node A will defer for a long enough duration enabling the node D to transmit a packet, as discussed in Section III.E. However, since node D is unaware that node A is gracefully deferring and the CW at node D may be very large due to its futilely retrying during the previous transmission between nodes A and B, the node D may not recognize this opportunity to transmit its packet. Therefore, the medium is unnecessary *idle* for a long time, and then node A may get control of the medium *again*, explaining the large throughput degrade and the short-term unfairness. In [1], the authors introduce the Request RTS (RRTS) frame at the receiver to help the sender to identify the *contention period*. We include this mechanism here. In particular, whenever a receiver (e.g., node C) *overhears* that some other receiver (i.e., node B) has sent out an *over-use notification*, the receiver (i.e., node C) should send out a RRTS to its *active* sender (i.e., node D). Upon receiving the RRTS, the sender (i.e., node D) will transmit immediately. We call this ATC-4, which extends ATC-3 by including this mechanism. Figure 8 and Table IV show that the ATC-4 greatly improves the fairness as well as the aggregate throughput compared to the ATC-3. Note that we have included the *Large-Col-EIFS* and *RRTS* mechanisms *only* in the simulations presented in *this* subsection, but not in others, since the pros and cons of these two mechanisms need further investigation.

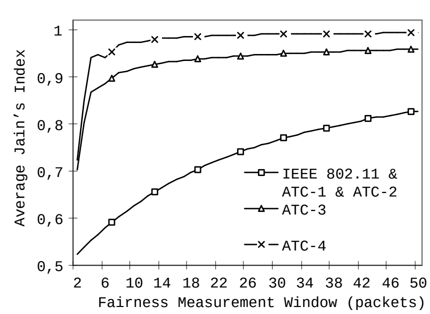

Fig. 8. Fairness Index under Topology of Figure 3

TABLE IV

THROUGHPUT UNDER THE TOPOLOGY OF FIGURE 3

Throughput (M bps)	IEEE 802.11	ATC-1	ATC-2	ATC-3	ATC-4
Aggregate	1.405	1.405	1.405	1.030	1.133

Topology of Figure 4: The Jain's index is presented in Figure 9. We notice that the ATCs do not improve the performance very drastically compared to the IEEE 802.11. The reason is that, in this topology, since the two flows can contend efficiently and the contention is not very high, the IEEE 802.11 itself delivers good fairness. The results presented in Table V show that the aggregate throughput improves slightly in the case of the ATC-2.

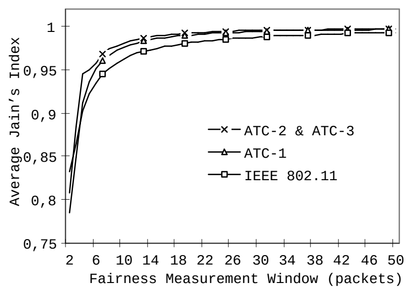

Fig. 9. Fairness Index under the Topology of Figure 4

TABLE V

THROUGHPUT UNDER THE TOPOLOGY OF FIGURE 4

Throughput (M bps)	IEEE 802.11	ATC-1	ATC-2	ATC-3
Aggregate	1.440	1.438	1.461	1.461

D. A complex topology

In this subsection, we consider a complex topology shown in Figure 10, which includes all the scenarios discussed so far. For instance, the nodes 0, 1 and 3 are *hidden* from each other. The nodes 2, 3, 5, and 7 form the topology shown in Figure 4, while the nodes 5, 7, 8, and 10 (as well as nodes 5,7,9 and 11; and nodes 8,9,10 and 11) form the *concealed information* topology shown in Figure 3.

It is clear that the flows f_1, f_2 and f_3 are in the same contention range and thus share a common medium (say C_1). Similarly, f_3, f_4 and f_5 share a common medium (say C_2), while f_5, f_6 and f_7 share another common medium (say C_3). Note that the flows f_3 and f_5 are contending in the two ranges (C_1 and C_2, and C_2 and C_3, respectively). Since the fairness is a measure of how evenly a *common* resource is distributed among the contenders, we will discuss the results *separately* for the three different mediums.

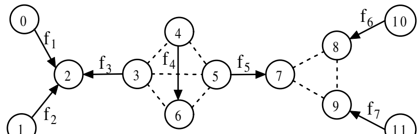

Fig. 10. A Complex Scenario

Figure 11-13 present the Jain's index for the three different mediums. It is easy to see that the ATCs substantially improve the fairness compared to that under IEEE 802.11. In the common medium C_1, the ATC-3 is *expected* to show the same fairness as ATC-2, since the receiver coordination mechanism does not play any *direct* role. However, the simulation results are somewhat different. The reason is that the receiver coordination mechanism of ATC-3 will affect the performance of flow f_5, which in turn affects flow f_3, and thus the ATC-3 in C_1 will provide different fairness in comparison to the ATC-2. Due to this *interaction* among different contention ranges, the ATC-1, -2, and -3 achieve different fairness in all the three ranges.

The throughput is presented in Table VI. We notice that, under IEEE 802.11, the flows f_3 and f_5 are almost starved, while the flow f_4 has a very large throughput (i.e., 1.211 Mbps). This is also indicated by the Jain's index (see the curve for IEEE 802.11 in Figure 12), as the index is almost equal to the *absolute unfairness* value (i.e., 1/N=1/3=0.33). The reason is as follows. Flow f_4 can get all the information about its contending flows (i.e., f_3 and f_5) and thus it can contend for the medium efficiently. On the contrary, flows f_3 and f_5 cannot contend efficiently, due to the hidden-terminal problem and the concealed information problem, respectively.

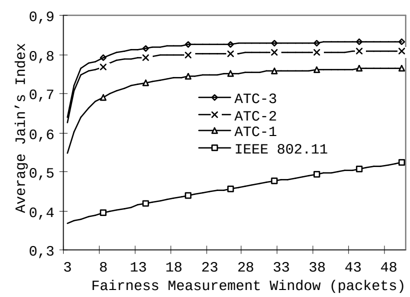

Fig. 11. Fairness Index for Common Medium C_1

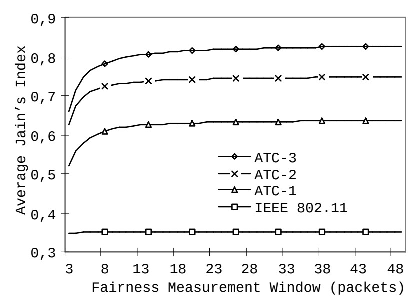

Fig. 12. Fairness Index for Common Medium C_2

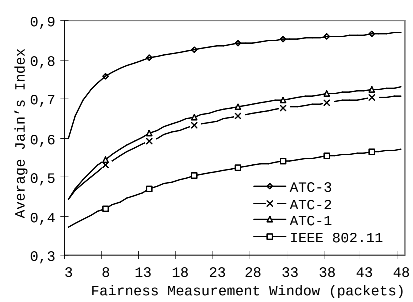

Fig. 13. Fairness Index for Common Medium C_3

In comparison to the situation under IEEE 802.11, in all the ATC schemes, these two flows (i.e., f_3 and f_5) get a certain amount of bandwidth, though it is still small compared to those of other flows. The question, why the throughputs of f_3 and

f_5 are small even in the ATC-3, can be answered as follows. The flow f_3 has to contend with *four* other flows (i.e., f_1, f_2, f_4, and f_5) and the flow f_5 also has to contend with *four* other flows (i.e., f_3, f_4, f_6, and f_7). On the contrary, each of the other flows has to contend with only *two* flows. Therefore, we argue that the distribution of the bandwidth in the ATC-3 is quite reasonable in such a complex topology (though whether it is reasonable or not depends upon how one defines the fairness). Note that the throughputs of f_6 and f_7 flows are less than those of f_1 and f_2, is due to the reason explained in Section IV.C.

The aggregate throughput under ATC-3 is about two-thirds of that under IEEE 802.11, showing the cost we have to pay for achieving the fairness as this topology involves the *worst-case* contention scenarios (e.g., hidden-terminal, concealed information, etc).

TABLE VI

THROUGHPUT UNDER THE COMPLEX TOPOLOGY

Throughput (Mbps)	IEEE 802.11	ATC-1	ATC-2	ATC-3
f_1	0.612	0.510	0.488	0.467
f_2	0.630	0.509	0.487	0.465
f_3	0.018	0.088	0.109	0.127
f_4	1.211	0.391	0.287	0.335
f_5	0.000	0.076	0.070	0.145
f_6	0.288	0.281	0.329	0.299
f_7	0.292	0.288	0.329	0.289
Aggregate	3.051	2.143	2.099	2.126

E. Verification of Estimation Algorithm

So far, using CBR traffic, we have seen that the ATCs substantially improve the fairness. From the simulation traces, we also found that the estimation of *n* converges very fast as indicated in Section III.B.3. 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 2 is used. 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 14 presents the results during a typical duration of 2 seconds. We notice that n_e is equal to *n* most of the time.

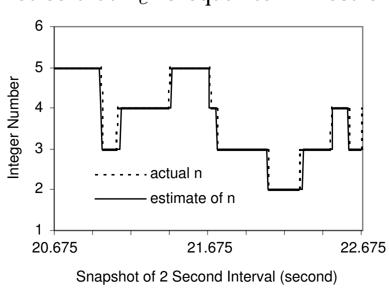

Fig. 14. Comparison between Estimate and Actual *n*

V. RELATED WORK

The fairness problem in random wireless MAC protocols was first highlighted in [1]. Recently, several work aim to develop a scheduler, which is *overlaid* on the top of the MAC layer, to address the unfairness problem due to the *location-dependent errors* (e.g., [13]) and the *location-dependent contention* in the multi-hop networks (e.g., [9], [12], and [16]). However, to implement the fair scheduling algorithm, global information and global synchronization is needed, which is difficult to obtain in the multi-hop wireless ad hoc networks. On the contrary, some other works aim to achieve fairness at the MAC layer itself. Our work belongs to this category. Most of the MAC fairness studies are based on a *distributed* CSMA/CA-based MAC protocol, which has two main components: collision avoidance (CA) and contention resolution (CR). In the literature, papers such as [2] mainly focus on the CA part to improve the fairness, while [6], [14], and [15] aim to achieve fairness by replacing the BEB with a new CR algorithm. In contrast, we keep the main feature of the BEB used in the IEEE 802.11. Also we mainly focus on the short-term fairness, whereas they mainly consider the long-term fairness.

The idea of adaptively adjusting the CW based on dynamic measurements is not new. By tuning the CW, [3] and [4] focus on improving capacity, rather than the fairness. To achieve fairness, [6] follows a similar approach as ours. In particular, they also dynamically estimate the sharing of the medium and then tune the CW. Our scheme is different from theirs in many aspects. For example, as they do not estimate *n*, the ϕ_i is always fixed at *0.5* irrespective of the number of active nodes, which leads to more collisions when *n* is large. Moreover, in [6] the CW is *directly* tuned, and there is nothing similar to our proposed early-reset mechanism. More importantly, the concealed information problem has not been addressed, which is common in the multi-hop ad hoc networks, demonstrating the importance of our receiver-coordination mechanism. Another related work [10] focuses on the topologies where *long-term* unfairness exists, and the *deterministic* approach has been used to implement a FIFO queue among the contending nodes, which results in substantial throughput degrade.

VI. CONCLUSIONS

In this paper, we have proposed the adaptive transmission control (ATC) algorithm, which is based on the dynamic estimation of the medium state. The simulation results show that the ATC greatly improve the fairness without unduly degrading the capacity utilization. Our ATC is novel in the sense that it is *simple*, *easy to implement*, *fully distributed*, *adaptive*, and *does not require any new control packets*. Our main contributions include: *(i)* identification of general problems that lead to short-term unfairness, e.g., hidden-terminal problem, high contention, and concealed information problem; *(ii)* proposal of a simple but efficient algorithm, which can dynamically estimate the number of *active* nodes within the contention range; *(iii)* proposal of the adaptive transmission control (ATC) fairness algorithm, which incorporate a suit of mechanisms, such as tuning CW, early-reset, and receiver coordination.

An issue that needs to be pursued in the future is to extend the ATC where the fair share is weighted according to the requirements of applications rather than being equal, and where the packet lengths are not equal.

REFERENCES

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

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

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

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

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

[6] Z. Fang, B. Bensaou, Y. Wang, ”Performance Evaluation of a Fair Backoff Algorithm for IEEE 802.11 DFWMAC,” ACM MOBIHOC, 2002.

[7] 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

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

[9] 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.

[10] —- ”Ordered Packet Scheduling in Wireless Ad Hoc Networks: Mechanisms and Performance Analysis,” in ACM MOBIHOC, 2002.

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

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

[13] S. Lu, T. Nandagopal and V. Bharghavan, ”A Wireless Fair Service Algorithm for Packet Cellular Networks,” ACM MOBICOM’98, Dallas, TX, October 1998.

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

[15] T. Ozugur, M. Naghshineh, P. Kermani, C. M Olsen, B. Rezvani, J.A. Copeland, ”Balanced Media Access Methods for Wireless Networks,” ACM MOBICOM, 1998.

[16] N.H. Vaidya, P. Bahl, ”Fair scheduling in broadcast environments,” Technical Report MSR-TR-99-61, Microsoft Research, Dec 1999.

[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