

Chapter 1

THROUGHPUT AND FAIRNESS OF COLLISION AVOIDANCE PROTOCOLS IN AD HOC NETWORKS *

J. J. Garcia-Luna-Aceves

*Department of Computer Engineering
University of California at Santa Cruz
Santa Cruz, CA 95064, U.S.A.
jj@cse.ucsc.edu*

Yu Wang

*Department of Computer Engineering
University of California at Santa Cruz
Santa Cruz, CA 95064, U.S.A.
ywang@cse.ucsc.edu*

Introduction

Wireless ad hoc networks have received increasing interest in recent years, because of their potential to be used in a variety of applications without the aid of any pre-existing network infrastructure.

Due to the scarce channel bandwidth available in ad hoc networks, the design of efficient and effective medium access control (MAC) protocols that regulate nodes' access to a shared channel has become the subject of active research in recent years. Many MAC protocols [1] have been proposed to mitigate the adverse effects of hidden terminals [6] through collision avoidance. Most collision avoidance schemes such as the carrier sense multiple access with collision avoidance (CSMA/CA) in the popular IEEE 802.11 MAC protocol [2] are sender-initiated, including an exchange of short request-to-send

*This work was supported in part by the Defense Advanced Research Projects Agency (DARPA) under Grant No. DAAD19-01-C-0026, the US Air Force/OSR under Grant No. F49620-00-1-0330 and the Jack Baskin Chair of Computer Engineering at UCSC.

(RTS) and clear-to-send (CTS) packets between a pair of sending and receiving nodes before the transmissions of the actual data packet and the optional acknowledgment packet.

In Section 1, we present an analytical modeling [7] to derive the saturation throughput of these sender-initiated collision avoidance protocols in multi-hop ad hoc networks with nodes randomly placed according to a two-dimensional Poisson distribution. We show that the sender-initiated collision-avoidance scheme achieves much higher throughput than the ideal carrier sense multiple access scheme with a separate channel for acknowledgments. More importantly, we show that the collision-avoidance scheme can accommodate much fewer competing nodes within a region in a network infested with hidden terminals than in a fully-connected network, if reasonable throughput is to be maintained. Simulations of the IEEE 802.11 MAC protocol and one of its variants validate the predictions made in the analysis.

The simulation results also reveal the fairness problem in IEEE 802.11 MAC protocol which refers to the severe throughput degradation of some nodes due to their unfavorable locations in the network and the commonly used binary exponential backoff (BEB) algorithm which always favors the node that last succeeds. This motivates the work presented in Section 2 in which we introduce a framework to address the fairness problem conclusively and propose a topology aware fair access (TAFE) scheme to realize the framework. Simulation results show that TAFE can solve the fairness problem in UDP-based applications with negligible degradation in throughput. It can also solve the notorious problem of the starvation of flows in TCP-based applications, while incurring only some throughput degradation. Hence, TAFE shows a much better overall tradeoff between throughput and fairness than other schemes previously proposed.

Section 3 concludes this chapter with directions for future work.

1. Performance of collision avoidance protocols

In Section 1.1, we present the analysis of the sender-initiated collision-avoidance scheme based on a four-way handshake and non-persistent carrier sensing, which can be also called the RTS/CTS-based scheme for the sake of simplicity. We first adopt a simple model in which nodes are randomly placed on a plane according to two-dimensional Poisson distribution with density λ . Varying λ has the effect of changing the congestion level within a region as well as the number of hidden terminals. In this model, it is also assumed that each node is ready to transmit independently in each time slot with probability p , where p is a protocol-dependent parameter. This model was first used by Takagi and Kleinrock [8] to derive the optimum transmission range of a node in a multi-hop wireless network, and was used subsequently by Wu and Varsh-

ney [9] to derive the throughputs of non-persistent CSMA and some variants of busy tone multiple access (BTMA) protocols [6]. Then we assume that both carrier sensing and collision avoidance work perfectly, that is, that nodes can accurately sense the channel busy or idle, and that the RTS/CTS scheme can avoid the transmission of data packets that collide with other packets at the receivers. The latter assumption can be called *perfect collision avoidance* and has been shown to be doable in the floor acquisition multiple access (FAMA) protocol [3]. Later we extend this model to take into account the possibility of data packets colliding with other transmissions, so that the model is also applicable to other MAC protocols, such as the popular IEEE 802.11 protocol, in which perfect collision avoidance is not strictly enforced.

In Section 1.2, we present numerical results from our analysis. We compare the performance of the sender-initiated collision avoidance scheme against the idealized non-persistent CSMA protocol in which a secondary channel is assumed to send acknowledgments in zero time and without collisions [6, 9], as the latter is the only protocol whose analysis for multi-hop ad hoc networks is available for comparison to date. It is shown that the RTS/CTS scheme can achieve far better throughput than the CSMA protocol, even when the overhead due to RTS/CTS exchange is high. The results illustrate the importance of enforcing collision avoidance in the RTS/CTS handshake.

However, the analytical results also indicate that the aggregate throughput of sender-initiated collision avoidance drops faster than that in a fully-connected network when the number of competing nodes within a region increases. This contrasts with conclusions drawn from the analysis of collision avoidance in fully-connected networks or networks with limited hidden terminals [3]. Our results show that hidden terminals degrade the performance of collision avoidance protocols beyond the basic effect of having a longer vulnerability period for RTSs. Hence, it follows that collision avoidance becomes more and more ineffective for a relatively crowded region with hidden terminals.

To validate the findings drawn from this analysis, in Section 1.3 we present simulations of the popular IEEE 802.11 MAC protocol. The simulation results clearly show that the IEEE 802.11 MAC protocol cannot ensure collision-free transmission of data packets, and that almost half of the data packets transmitted cannot be acknowledged due to collisions, even when the number of competing nodes in a neighborhood is only eight! However, the performance of the simulated IEEE 802.11 MAC protocol correlates well with what is predicted in the extended analysis, which takes into account the effect of data packet collisions and is used for the case when the number of competing nodes in a region is small. When the number of competing nodes in a region increases, the performance gap between IEEE 802.11 and the analysis decreases, which validates the statement that even a perfect collision-avoidance protocol loses its

effectiveness gradually due to the random nature of the channel access and the limited information available to competing nodes.

The simulation results for the IEEE 802.11 protocol also show a larger variation in throughput than the predicted performance from the analytical model, which is due to its inherent fairness problems which motivates the second part of the work reported in this chapter.

1.1 Approximate Analysis

In this section, we derive the approximate throughput of a perfect collision avoidance protocol. In our network model, nodes are two-dimensionally Poisson distributed over a plane with density λ , i.e., the probability $p(i, S)$ of finding i nodes in an area of S is given by:

$$p(i, S) = \frac{(\lambda S)^i}{i!} e^{-\lambda S}.$$

Assume that each node has the same transmission and receiving range of R , and denote by N the average number of nodes within a circular region of radius R ; therefore, we have $N = \lambda \pi R^2$.

To simplify our analysis, we assume that nodes operate in time-slotted mode. As prior results for CSMA and collision-avoidance protocols show [6], the performance of MAC protocols based on carrier sensing is much the same as the performance of their time-slotted counterparts in which the length of a time slot equals one propagation delay and the propagation delay is much smaller than the transmission time of data packets.

The length of each time slot is denoted by τ . Note that τ is not just the propagation delay, because it also includes the overhead due to the transmit-to-receive turn-around time, carrier sensing delay and processing time. In effect, τ represents the time required for all the nodes within the transmission range of a node to know the event that occurred τ seconds ago. The transmission times of RTS, CTS, data, and ACK packets are normalized with regard to τ , and are denoted by l_{rts} , l_{cts} , l_{data} , and l_{ack} , respectively. Thus, τ is also equivalent to 1 in later derivations. For the sake of simplicity, we also assume that all packet transmission times are multiples of the length of a time-slot.

We derive the protocol's throughput based on the heavy-traffic assumption, i.e., a node always has a packet in its buffer to be sent and the destination is chosen randomly from one of its neighbors. This is a fair assumption in ad hoc networks in which nodes are sending data and signaling packets continually. We also assume that a node is ready to transmit with probability p and not ready with probability $1 - p$. Here p is a protocol-specific parameter that is slot independent. At the level of individual nodes, the probability of being ready to transmit may vary from time slot to slot, depending on the current states of both the channel and the node. However, because we are interested

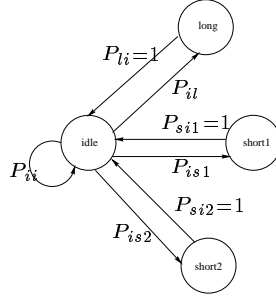


Figure 1.1. Markov chain model for the channel around a node

in deriving the average performance metrics instead of instantaneous or short-term metrics, the assumption of a fixed probability p may be considered as an averaged quantity that can still reasonably approximate the factual burstiness from a long-term point of view. In fact, this assumption is necessary to make the theoretical modeling tractable and has been extensively applied before [–5, 10, 8]. For example, this model was used by Takagi and Kleinrock [8] to derive the optimal transmission range of a node in a multi-hop wireless network, and was used subsequently by Wu and Varshney [9] to derive the throughput of non-persistent CSMA and some variants of busy tone multiple access (BTMA) protocols [6].

It should also be noted that, even when a node is ready to transmit, it may transmit or not in the slot, depending on the collision avoidance and resolution schemes being used, as well as the channel’s current state. Thus, we are more interested in the probability that a node transmits in a time slot, which is denoted by p' . Similar to the reasoning presented for p , we also assume that p' is independent at any time slot to make the analysis tractable. Given this simplification, p' can be defined to be

$$p' = p \cdot \text{Prob.}\{\text{Channel is sensed idle in a slot}\} \\ \approx p \cdot \Pi_I$$

where Π_I is the limiting probability that the channel is in idle state, which we derive subsequently.

We are not interested in the exact relationship between p and p' , and it is enough to obtain the range of values that p' can take, because the throughput of these protocols is mostly influenced by p' . To derive the rough relationship between p and p' , we set up a channel model that includes two key simplifying assumptions.

First, we model the channel as a circular region in which there are some nodes. The nodes within the region can communicate with each other while they have weak interactions with nodes outside the region. *Weak interaction*

means that the decision of inner nodes to transmit, defer and back off is almost not affected by that of outer nodes and vice versa. Considering that nodes do not exchange status information explicitly (e.g., either defer due to collision avoidance or back off due to collision resolution), this assumption is reasonable and helps to simplify the model considerably. Thus, the channel's status is only decided by the successful and failed transmissions within the region.

Second, we still consider the failed handshakes initiated by nodes within the region to outside nodes, because this has a direct effect on the channel's usability for other nodes within the region. Though the radius of the circular region R' is unknown, it falls between $R/2$ and $2R$. This follows from noting that the maximal radius of a circular region in which all nodes are guaranteed to hear one another equals $R' = R/2$, and all the direct neighbors and hidden nodes are included into the region when $R' = 2R$. Thus, we obtain $R' = \alpha R$ where $0.5 \leq \alpha \leq 2$, and α needs to be estimated.

With the above assumptions, the channel can be modeled by a four-state Markov chain illustrated in Figure 1.1. The significance of the states of this Markov chain is the following:

- *Idle* is the state when the channel around node x is sensed idle, and obviously its duration is τ .
- *Long* is the state when a successful four-way handshake is done. For simplicity, we assume that the channel is in effect busy for the duration of the whole handshake, thus the busy time T_{long} is

$$\begin{aligned} T_{long} &= l_{rts} + \tau + l_{cts} + \tau + l_{data} + \tau + l_{ack} + \tau \\ &= l_{rts} + l_{cts} + l_{data} + l_{ack} + 4\tau. \end{aligned}$$

- *Short1* is the state when multiple nodes around the channel transmit RTS packets during the same time slot and their transmissions collide. The busy time of the channel T_{short1} is therefore

$$T_{short1} = l_{rts} + \tau.$$

- *Short2* is the state when one node around the channel initiates a failed handshake with a node outside the region. Even though a CTS packet may not be sent due to the collision of the sending node's RTS packet with other packets originated from nodes outside the region or due to the deferring of the receiving node to other nodes, those nodes overhearing the RTS as well as the sending node do not know if the handshake is successfully continued, until the time required for receiving a CTS packet elapses. Therefore the channel is in effect busy, i.e., unusable for all the nodes sharing the channel, for the time stated below:

$$\begin{aligned} T_{short2} &= l_{rts} + \tau + l_{cts} + \tau \\ &= l_{rts} + l_{cts} + 2\tau. \end{aligned}$$

Now we proceed to calculate the transition probabilities of the Markov chain.

In most collision avoidance schemes with non-persistent carrier sensing, no node is allowed to transmit immediately after the channel becomes idle, thus the transition probabilities from *long* to *idle*, from *short1* to *idle* and from *short2* to *idle* are all 1.

According to the Poisson distribution of the nodes, the probability of having i nodes within the receiving range R of x is $e^{-N} N^i / i!$, where $N = \lambda \pi R^2$. Therefore, the mean number of nodes that belong to the shared channel is $M = \lambda \pi R'^2 = \alpha^2 N$. Assuming that each node transmits independently, the probability that none of them transmits is $(1 - p')^i$, where $(1 - p')$ is the probability that a node does not transmit in a time slot. Because the transition probability P_{ii} from *idle* to *idle* is the probability that none of the neighboring nodes of x transmits in this slot, P_{ii} is given by

$$\begin{aligned} P_{ii} &= \sum_{i=0}^{\infty} (1 - p')^i \frac{M^i}{i!} e^{-M} \\ &= \sum_{i=0}^{\infty} \frac{[(1 - p')M]^i}{i!} e^{-(1-p')M} \cdot e^{-p'M} = e^{-p'M}. \end{aligned}$$

We average the probabilities over the number of interfering nodes in a region because of two reasons. First, it is much more tractable than the approach that conditions on the number of nodes, calculates the desired quantities, and then uses the Poisson distribution to obtain the average. Second, in our simulation experiments, we fix the number of competing nodes in a region (which is N) and then vary the location of the nodes to approximate the Poisson distribution, which is configurationally closer to our analytical model; the alternative would be to generate 2, 3, 4, . . . nodes within one region, get the throughput for the individual configuration and then calculate the average, which is not practical.

Next we need to calculate the transition probability P_{il} from *idle* to *long*. If there are i nodes around node x , for such a transition to happen, one and only one node should be able to complete one successful four-way handshake while other nodes do not transmit. Let p_s denote the probability that a node begins a successful four-way handshake at each slot, we can then calculate P_{il} as follows:

$$\begin{aligned} P_{il} &= \sum_{i=1}^{\infty} i p_s (1 - p')^{i-1} \frac{M^i}{i!} e^{-M} \\ &= \sum_{i=1}^{\infty} p_s (1 - p')^{i-1} \frac{M^{i-1}}{(i-1)!} M e^{-M} \\ &= p_s M \sum_{i=0}^{\infty} \frac{[M(1 - p')]^i}{i!} e^{-M(1-p'+p')} \\ &= p_s M e^{-p'M}. \end{aligned}$$

To obtain the above result, we use the fact that the distribution of the number of nodes within R' does not depend on the existence of node x , because of the memoryless property of the Poisson distribution. Up to this point, p_s is still an unknown quantity that we derive subsequently.

The transition probability from *idle* to *short1* is the probability that more than one node transmit RTS packets in the same slot; therefore, P_{is1} can be calculated as follows:

$$\begin{aligned} P_{is1} &= \sum_{i=2}^{\infty} [1 - (1-p')^i - ip'(1-p')^{i-1}] \frac{M^i}{i!} e^{-M} \\ &= 1 - (1 + Mp')e^{-p'M}. \end{aligned}$$

Having calculated P_{ii} , P_{il} and P_{is1} , we can calculate P_{is2} , the transition probability from *idle* to *short2*

$$\begin{aligned} P_{is2} &= 1 - P_{ii} - P_{il} - P_{is1} \\ &= 1 - e^{-p'M} - p_s M e^{-p'M} - (1 - (1 + Mp')e^{-p'M}) \\ &= (p' - p_s) M e^{-p'M}. \end{aligned}$$

Let π_i , π_l , π_{s1} and π_{s2} denote the steady-state probabilities of states *idle*, *long*, *short1* and *short2*, respectively. From Figure 1.1, we have

$$\begin{aligned} \pi_i P_{ii} + \pi_l + \pi_{s1} + \pi_{s2} &= \pi_i \\ \pi_i P_{ii} + 1 - \pi_i &= \pi_i \\ \pi_i &= \frac{1}{2 - P_{ii}} = \frac{1}{2 - e^{-p'M}}. \end{aligned}$$

The limiting probability Π_I , i.e., the long run probability that the channel around node x is found idle, can be obtained by:

$$\Pi_I = \frac{\pi_i T_{idle}}{\pi_i T_{idle} + \pi_l T_{long} + \pi_{s1} T_{short1} + \pi_{s2} T_{short2}}.$$

Noting that $\pi_i P_{il} = \pi_l$, $\pi_i P_{is1} = \pi_{s1}$ and $\pi_i P_{is2} = \pi_{s2}$, we obtain

$$\begin{aligned} \Pi_I &= \frac{\pi_i T_{idle}}{\pi_i T_{idle} + \pi_i P_{il} T_{long} + \pi_i P_{is1} T_{short1} + \pi_i P_{is2} T_{short2}} \\ &= \frac{T_{idle}}{T_{idle} + P_{il} T_{long} + P_{is1} T_{short1} + P_{is2} T_{short2}}. \end{aligned}$$

The relationship between p' and p is then:

$$\begin{aligned} p' &= \frac{p T_{idle}}{T_{idle} + P_{il} T_{long} + P_{is1} T_{short1} + P_{is2} T_{short2}} \\ &= \frac{p T_{idle}}{T_{idle} + p_s M e^{-p'M} T_{long} + (1 - (1 + p'M)e^{-p'M}) T_{short1} + \dots} \\ &\quad \dots + (p' - p_s) M e^{-p'M} T_{short2} \end{aligned} \tag{1.1}$$

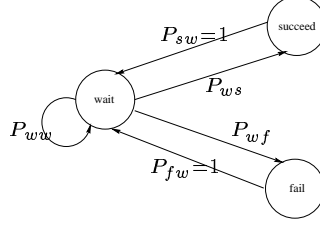


Figure 1.2. Markov chain model for a node

In the above equation, the probability that a node x starts successfully a four-way handshake in a time slot, p_s , is yet to be determined.

The states of a node x can be modeled by a three-state Markov chain, which is shown in Figure 1.2.

In Figure 1.2, *wait* is the state when the node defers for other nodes or backs off, *succeed* is the state when the node can complete a successful four-way handshake with other nodes, and *fail* is the state when the node initiates an unsuccessful handshake. For simplicity, we regard *succeed* and *fail* as the states when two different kinds of *virtual* packets are transmitted and their lengths are:

$$T_{succeed} = T_{long} = l_{rts} + l_{cts} + l_{data} + l_{ack} + 4\tau$$

$$T_{fail} = T_{short2} = l_{rts} + l_{cts} + 2\tau.$$

Obviously, the duration of a node in *wait* state T_{wait} is τ .

Because by assumption collision avoidance is enforced at each node, no node is allowed to transmit data packets continuously; therefore, the transition probabilities from *succeed* to *wait* and from *fail* to *wait* are both one.

To derive the transition probability P_{ws} from *wait* to *succeed*, we need to calculate the probability $P_{ws}(r)$ that node x successfully initiates a four-way handshake with node y at a given time slot when they are at a distance r apart. Before calculating $P_{ws}(r)$, we define $B(r)$ to be the area that is in the hearing region of node y but outside the hearing region of node x , i.e., the interfering region “hidden” from node x as the shaded area shown in Figure 1.3. $B(r)$ has been shown in [8] to be:

$$B(r) = \pi R^2 - 2R^2 q\left(\frac{r}{2R}\right) \quad (1.2)$$

where $q(t) = \arccos(t) - t\sqrt{1-t^2}$.

Then $P_{ws}(r)$ can be calculated as:

$$P_{ws}(r) = P_1 \cdot P_2 \cdot P_3 \cdot P_4(r)$$

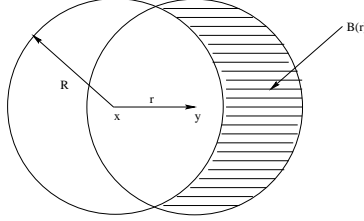


Figure 1.3. Illustration of “hidden” area

where

$$P_1 = \text{Prob.}\{x \text{ transmits in a slot}\},$$

$$P_2 = \text{Prob.}\{y \text{ does not transmit in the time slot}\},$$

$$P_3 = \text{Prob.}\{\text{none of the terminals within } R \text{ of } x \text{ transmits in the same slot}\},$$

$$P_4(r) = \text{Prob.}\{\text{none of the terminals in } B(r) \text{ transmits for } (2l_{rts} + 1) \text{ slots} \mid r\}.$$

The reason for the last term is that the vulnerable period for an RTS is only $2l_{rts} + 1$, and once the RTS is received successfully by the receiving node (which can then start sending the CTS), the probability of further collisions is assumed to be negligibly small.

Obviously, $P_1 = p'$ and $P_2 = (1 - p')$. On the other hand, P_3 can be obtained by

$$\begin{aligned} P_3 &= \sum_{i=0}^{\infty} (1 - p')^i \frac{(\lambda\pi R^2)^i}{i!} e^{-\lambda\pi R^2} \\ &= \sum_{i=0}^{\infty} (1 - p')^i \frac{N^i}{i!} e^{-N} \\ &= e^{-p'N}. \end{aligned}$$

Similarly, the probability that none of the terminals in $B(r)$ transmits in a time slot is given by

$$\begin{aligned} p_4(r) &= \sum_{i=0}^{\infty} (1 - p')^i \frac{(\lambda B(r))^i}{i!} e^{-\lambda B(r)} \\ &= e^{-p'\lambda B(r)}. \end{aligned}$$

Hence, $P_4(r)$ can be expressed as

$$\begin{aligned} P_4(r) &= (p_4(r))^{2l_{rts} + 1} \\ &= e^{-p'\lambda B(r)(2l_{rts} + 1)}. \end{aligned}$$

Given that each sending node chooses any one of its neighbors with equal probability and that the average number of nodes within a region of radius r is

proportional to r^2 , the probability density function of the distance r between node x and y is

$$f(r) = 2r, \quad 0 < r < 1.$$

where we have normalized r with regard to R by setting $R = 1$.

Now we can calculate P_{ws} as follows:

$$\begin{aligned} P_{ws} &= \int_0^1 2r P_{ws}(r) dr \\ &= 2p'(1-p')e^{-p'N} \int_0^1 r e^{-p'\lambda B(r)(2l_{rts}+1)} dr \\ &= 2p'(1-p')e^{-p'N} \int_0^1 r e^{-p'N[1-2q(r/2)/\pi](2l_{rts}+1)} dr. \end{aligned} \quad (1.3)$$

From the Markov chain shown in Figure 1.2, the transition probability P_{ww} that node x continues to stay in *wait* state in a slot is just $(1-p')e^{-p'N}$, i.e., node x does not initiate any transmission and there is no node around it initiating a transmission. Let π_s , π_w and π_f denote the steady-state probability of state *succeed*, *wait* and *fail*, respectively. From Figure 1.2, we have

$$\begin{aligned} \pi_w P_{ww} + \pi_s + \pi_f &= \pi_w \\ \pi_w P_{ww} + 1 - \pi_w &= \pi_w \\ \pi_w &= \frac{1}{2 - P_{ww}} = \frac{1}{2 - (1-p')e^{-p'N}}. \end{aligned}$$

Therefore, the steady-state probability of state *succeed*, π_s , can be calculated as:

$$\pi_s = \pi_w P_{ws} = \frac{P_{ws}}{2 - (1-p')e^{-p'N}} = p_s. \quad (1.4)$$

Equation (1.4) points out the fact that π_s is just the previous unknown quantity p_s in Equation (1.1). Combining Equations (1.1), (1.3) and (1.4) together, we get a complex relationship between p and p' . However, given p , p' can be computed easily with numerical methods.

Accordingly, the throughput Th is:

$$\begin{aligned} Th &= \frac{\pi_s \cdot l_{data}}{\pi_w T_w + \pi_s T_s + \pi_f T_f} \\ &= \frac{l_{data} \pi_s}{\tau \pi_w + (l_{rts} + l_{cts} + l_{data} + l_{ack} + 4\tau) \pi_s} \cdots \\ &\quad \cdots \frac{\pi_s}{(l_{rts} + l_{cts} + 2\tau)(1 - \pi_w - \pi_s)}. \end{aligned} \quad (1.5)$$

From the formula used to calculate throughput, we can see that π_s and π_w , from which throughput is derived, are largely dependent on p' and not on p , which is the basis for our simplification of the modeling of the channel presented earlier.

To apply our analysis to MAC protocols in which perfect collision avoidance is not enforced, e.g., the IEEE 802.11 MAC protocol, we propose a simple though not rigorous extension of the analysis. We can add another state to the Markov chain for the node model (ref. Figure 1.2) whose duration is $l_{rts} + l_{cts} + l_{data} + 3\tau$. This is a *pseudo-succeed* state in which an RTS-CTS-data handshake takes place without acknowledgment coming back due to collisions, i.e., it is a state derived from the *succeed* state of the perfect collision avoidance protocol. We use an “imperfection factor” β to model the deviatory behavior of the protocol, given that different MAC protocols may have different values of β . The transition probability from *wait* to the *pseudo-succeed* state is then βP_{ws} , and the transition probability from *wait* to *succeed* is $(1 - \beta)P_{ws}$. Hence, the modified formula for throughput is simply:

$$\begin{aligned} Th = & (1 - \beta)l_{data}\pi_s[\tau\pi_w + (l_{rts} + l_{cts} + l_{data} + l_{ack} + 4\tau)(1 - \beta)\pi_s \\ & + (l_{rts} + l_{cts} + 2\tau)(1 - \pi_w - \pi_s) + (l_{rts} + l_{cts} + l_{data} + 3\tau)\beta\pi_s]^{-1} \end{aligned} \quad (1.6)$$

When the deviatory factor β equals zero, Equation (1.6) is reduced to Equation (1.5).

1.2 Numerical Results

In this section, we compare the throughput of the RTS/CTS scheme with a non-persistent CSMA protocol in which there is a separate channel over which acknowledgments are sent in zero time and without collisions. The performance of the latter protocol in multi-hop networks has been analyzed by Wu and Varshney [9] and we should note that, in practice, the performance of the CSMA protocol would be worse as both data packets and acknowledgments are transmitted in the same channel.

We present results when either relatively large data packets or relatively small data packets are sent. Let τ denote the duration of one time slot. RTS, CTS and ACK packets last 5τ . As to the size of data packets, we consider two cases. One case corresponds to a data packet that is much larger than the aggregate size of RTS, CTS and ACK packets. The other case corresponds to a data packet being only slightly larger than the aggregate size of RTS, CTS and ACK packets. In the latter case, which models networks in which radios have long turn-around times and data packets are short, it is doubtful whether a collision avoidance scheme should be employed at all, because it represents excessive overhead.

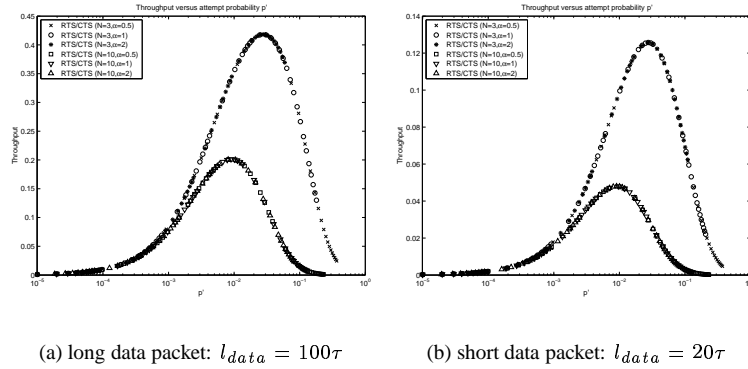


Figure 1.4. α 's influence ($l_{rts} = l_{cts} = l_{ack} = 5\tau$)

We first calculate throughput with different values of α , which we define as the ratio between the circular region including nodes affected by an RTS/CTS handshake and the largest possible circular region in which nodes are guaranteed to be connected with one another. We find that, though the relationship between the ready probability p and transmission-attempt probability p' under different values of α might be somewhat different, the throughput is largely unaffected by α , which is shown in Figure 1.4.¹ In Figure 1.4, N is the average number of nodes that compete against one another to access the shared channel. Thus, the burden of estimating α is relieved in our model, and we can focus on the case in which $\alpha = 1$ thereafter. However, as a side effect of not knowing the actual α that should be used, the relationship between p' and throughput may not agree with the simulations. However, for our purposes this is not a problem, because we are interested in the saturated throughput only.

Figure 1.5 compares the throughput of collision avoidance against that of CSMA with different values of N and data packet lengths, and we can make the following observations from the above results.

When data packet is long, the throughput of CSMA is very low, even for the case in which only $N = 3$ nodes are competing for the shared channel. By comparison, the RTS/CTS scheme can achieve much higher throughput, even when the average number of competing nodes is 10. The reason is simple, the larger a data packet is, the worse the impact of hidden terminals is for that packet in CSMA, because the vulnerability period becomes twice the length of the data packet. With collision avoidance, the vulnerability period of a handshake is independent of the length of data packets, and in the worse case, equals twice the length of an RTS. When a data packet is not very long and the overhead of the collision avoidance and handshake seems to be rather high, collision avoidance can still achieve marginally better throughput than CSMA.

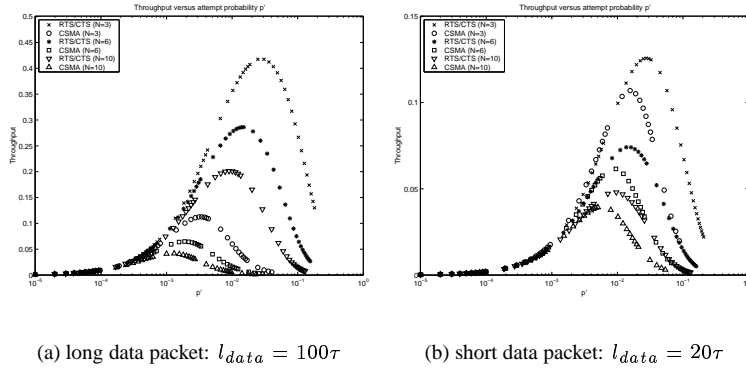


Figure 1.5. Throughput comparison ($l_{rts} = l_{cts} = l_{ack} = 5\tau$)

We need to emphasize that the performance of the actual CSMA protocol would be much worse than the idealized model we have used for comparison purposes, because of the effect of acknowledgments.

Despite the advantage of collision avoidance, its throughput still degrades rapidly with the increase of N . This is also evident for low values of p' as shown in Figure 1.5. This is due the fact that nodes are spending much more time on collision avoidance and backoff. When N increases, p' decreases much slower to achieve optimum throughput, which already decreases. This shows that collision avoidance becomes more and more ineffective when the number of competing nodes within a region increases, even though these nodes are quite “polite” in their access to the shared channel. This is also different from a fully-connected network, in which the maximum throughput is largely indifferent to the number of nodes within a region [11].

Our results also reveal that hidden terminals degrade the performance of collision avoidance protocols beyond the basic effect of having a longer vulnerability period for RTSs. There is one dilemma here. On the one hand, it is very difficult to get all the competing nodes around one node coordinated well by probabilistic methods such as randomized backoff. Here the competing nodes refer to both one-hop and two-hop neighbors² of the node. In actual MAC protocols, the collisions of data packets may still occur and throughput degrades with increasing numbers of neighbors. On the other hand, even if all the competing nodes of one node defer their access for the node, the possible spatial reuse in multi-hop networks is greatly reduced and hence the maximum achievable throughput is reduced. This dilemma leads to the scalability problem of contention-based MAC protocols that occurs much earlier than people might expect, as the throughput is already quite meager when the average of competing nodes within a region (N) is only ten.

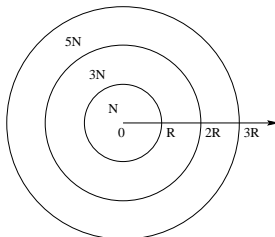


Figure 1.6. Network Model Illustration

1.3 Simulation Results

The numerical results in the previous section show that an RTS/CTS based access scheme outperforms CSMA, even when the overhead of RTS/CTS packets is comparable to the data packets to be transmitted if perfect collision avoidance can be achieved. In this section, we investigate the performance of the popular IEEE 802.11 DFWMAC protocol to validate the predictions made in the analysis.

We use GloMoSim 2.0 [12] as the network simulator. Direct sequence spread spectrum (DSSS) parameters are used throughout the simulations, which are shown in Table 1.1. The raw channel bit rate is 2Mbps. We use a uniform distribution to approximate the Poisson distribution used in our analytical model, because the latter is mainly used to facilitate our derivation of analytical results. In addition, it is simply impractical to generate 2, 3, 4, ... nodes within one region, get the throughput for the individual configuration and then calculate the average like what is required in the analytical model. In the network model used simulations, we place nodes in concentric circles or rings as illustrated in Figure 1.6. That is, given that a node's transmitting and receiving range is R and that there are on average N nodes within this circular region, we place N nodes in a circle of radius R , subject to a uniform distribution. Because there are on average 2^2N nodes within a circle of radius $2R$, we place $2^2N - N = 3N$ nodes outside the previous circle of radius R but inside the concentric circle of radius $2R$, i.e., the ring with radii R and $2R$, subject to the same uniform distribution. Then $3^2N - 2^2N = 5N$ nodes can be placed in an outer ring with radii $2R$ and $3R$.

Because it is impossible to generate the infinite network we assumed in our analysis in simulations, we just focus our attention on the performance of the innermost N nodes. Another reason is that it is more appropriate to investigate the performance of MAC schemes in a local neighborhood, rather than in the whole network, because totaling and averaging performance metrics such as throughput and delay with regard to all the nodes both in the center and at the edge of a network may lead to some askew results. For example, nodes at the

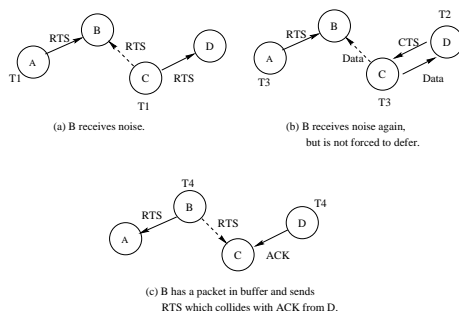


Figure 1.7. Example of collisions with data packets in the IEEE 802.11 MAC Protocol

edge may have exceedingly high throughput due to much less contention and including them in the calculation would lead to higher than usual throughput. In our experiments, we find that nodes that are outside the concentric circles of radius $3R$ almost have no influence on the throughput of the innermost N nodes, i.e., boundary effects can be safely ignored when the circular network's radius is $3R$. Accordingly, we present only the results for a circular network of radius $3R$.

The backoff timer in the IEEE 802.11 MAC protocol is drawn from a uniform distribution whose upper bound varies according to the estimated contention level, i.e., a modified binary exponential backoff. Thus, p' takes on dynamic values rather than what we have assumed in the analytical model. Accordingly, we expect that the IEEE 802.11 MAC protocol will operate in a region, while our analysis gives only average performance. In addition, even in network topologies that satisfy the same uniform distribution, we can still get quite different results, which will be shown later.

As we have stated, the IEEE 802.11 MAC protocol cannot ensure collision-free transmission of data packets, even under the assumption of perfect carrier sensing and collision avoidance. There are two reasons for this. One is that the length of a CTS is shorter than that of an RTS, which has been shown to prevent some hidden nodes from backing off [3]. The other reason is that, when a node senses carrier in its surroundings, it does not defer access to the channel for a definite time (which is implicit in other protocols [3]) after the channel is clear. When the interfering node perceives the channel idle and a packet from the upper layer happens to arrive in its buffer, it may transmit immediately after the channel is idle for a DIFS (Distributed InterFrame Space) time, while in fact a data packet transmission may still be going on between another two nodes and collision will occur! This can be illustrated by the simple example shown in Figure 1.7.

In our simulation, each node has a constant-bit-rate (CBR) traffic generator with data packet size of 1460 bytes, and one of its neighbors is randomly chosen

Table 1.1. IEEE 802.11 protocol configuration parameters

<i>RTS</i>	<i>CTS</i>	<i>data</i>	<i>ACK</i>	<i>DIFS</i>	<i>SIFS</i>
20-byte	14-byte	1460-byte	14-byte	50 μ sec	10 μ sec
<i>contention window</i>		<i>slot time</i>	<i>sync. time</i>	<i>prop. delay</i>	
31–1023		20 μ sec	192 μ sec	1 μ sec	

Table 1.2. Equivalent configuration parameters for analytical model

	τ	l_{rts}	l_{cts}, l_{ack}	l_{data}
actual time	21 μ sec	272 μ sec	248 μ sec	6032 μ sec
normalized	1	13	12	287

as the destination for each packet generated. All nodes are always backlogged. Considering the physical layer’s synchronization time as well as propagation delay used in the simulation, the effective packet transmission times are shown in Table 1.1. For comparison purposes, we map these simulational parameters to equivalent parameters in our analytical model and they are shown in Table 1.2.

We run both analytical and simulation programs with $N = 3, 5$ and 8 . Though we have not tried to characterize how the performance of the IEEE 802.11 MAC protocol is distributed in the region of values taken by p' , we do have generated 50 random topologies that satisfy the uniform distribution and then get an average transmission probability and throughput for the N nodes in the innermost circle of radius R for each configuration. The results are shown in Figure 1.8, in which the centers of rectangles are the mean values of p' and throughput and their half widths and half heights are the variance of p' and throughput, respectively. These rectangles roughly describe the operating regions of IEEE 802.11 MAC protocol with the configurations we are using.

Figure 1.8 clearly shows that, IEEE 802.11 cannot achieve the performance predicted in the analysis of correct collision avoidance, but may well outperform the analysis with the same p' for some configurations, especially when N is small. On first thought, it may seem contrary to intuition, given that IEEE 802.11 cannot ensure collision-free data packet transmissions and should always perform worse than analysis results. In fact, the exceedingly high throughput is largely due to the unfairness of the binary exponential backoff (BEB) used in IEEE 802.11. In BEB, a node that just succeeds in sending a data packet

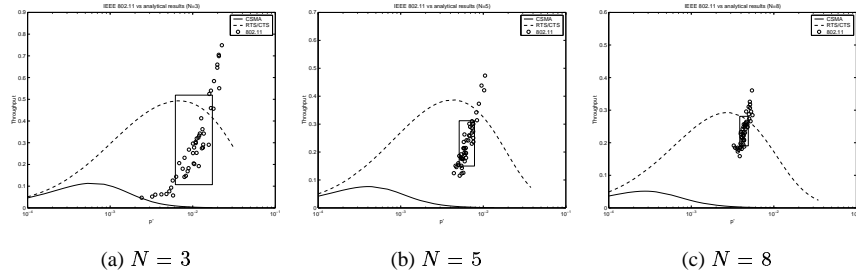


Figure 1.8. Performance comparison of IEEE 802.11 with analytical results

Table 1.3. Percentage of ACK timeout in BEB scheme

	N = 3	N = 5	N = 8
mean	0.29	0.39	0.44
std	0.17	0.10	0.06

resets its contention window to the minimum value, through which it may gain access to the channel again much earlier than other surrounding nodes. Thus, a node may monopolize the channel for a very long time during which there is no contention loss and throughput can be very high for a particular node, while other nodes suffer starvation. We also find that when N increases, the variance of p' and throughput becomes smaller. Thus, the fairness problem is less severe when there are more nodes competing in a shared channel.

Given that the IEEE 802.11 MAC protocol cannot ensure that data packets are transmitted free of collisions, its throughput can deviate much from what is predicted in the analysis. To demonstrate this, we also collect statistics about the number of transmitted RTS packets that will lead to ACK timeout due to collision of data packets as well as the total number of transmitted RTS packets that can lead to either an incomplete RTS-CTS-data handshake or a successful four-way handshake. Then we calculate the ratio of these two numbers and tabulate the results in Table 1.3. This table clearly shows that much of the precious channel resource is wasted in sending data packets that cannot be successfully delivered.

A close observation of Figure 1.8 also reveals that, the gap in maximum throughput between analytical and simulation results decreases when N increases. This can be explained as follows. When the number of direct competing nodes N increases, the number of indirect competing nodes (hidden

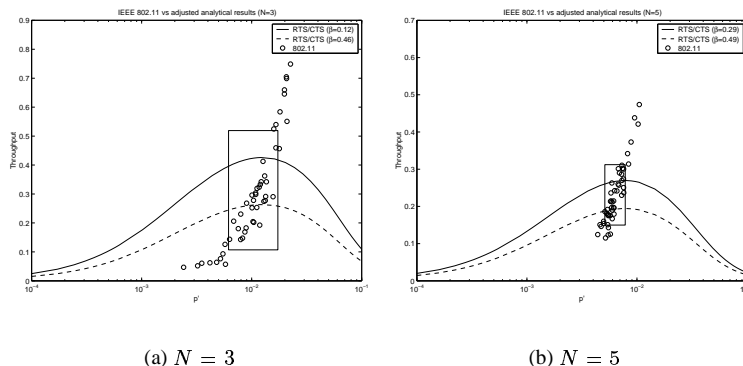


Figure 1.9. Performance comparison of IEEE 802.11 with adjusted analytical results

terminals, $3N$ on average) also increases, which makes nodes implementing a perfect collision avoidance protocol spend much more time in deferring and backing off to coordinate with both one-hop and two-hop competing nodes to avoid collisions. Therefore, much of the gain of perfect collision avoidance is lost and possible spatial reuse is also reduced in congested area, which makes a perfect collision avoidance protocol work only marginally better than an imperfect one. This observation could not be predicted from previous analytical models or simulations focusing on fully-connected networks or networks with only a limited number of hidden terminals [9, 11, 10, 13].

The percentage shown in Table 1.3 is in fact the β in our extended analysis to explain the deviatory behavior of MAC protocols that do not have perfect collision avoidance. Using these values, we compare the performance of the IEEE 802.11 protocol with that of the adjusted analysis obtained from Equation (1.6), and show the results in Figure 1.9. In Figure 1.9, we only show the results for small values of N as it is not quite meaningful to do the adjustment for large values of N due the reason stated above. Figure 1.9 shows that the extended analysis is a rather good approximation of the actual performance of the IEEE 802.11 protocol though the latter has larger variation in throughput (possibly due to its inherent fairness problems).

2. Framework and Mechanisms for Fair Access in IEEE 802.11

As we have stated, the fairness problem is due to some nodes' unfavorable location in the network and the commonly used binary exponential backoff (BEB) aggravates this problem. The fairness problem is not new and there is already some work done on it. The work so far can be roughly categorized

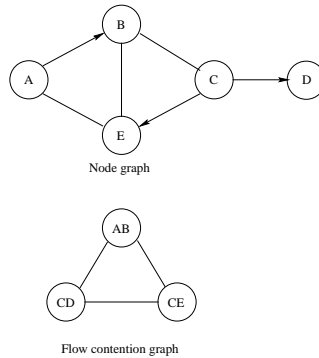


Figure 1.10. A simple network: node graph and flow contention graph

into two classes. In the first class, the goal is to achieve max-min fairness [] by reducing the ratio between maximum throughput and minimum throughput of flows, either at a node's level or at a flow's level. In the second class, the approach used in fair queuing for wireline networks is adapted to multi-hop ad hoc networks taking into account location dependent contention [] and flow contention graphs are used extensively in the schemes in the second class to model the contention among nodes. Figure 1.10 shows an example of how this is done. Any two flows with adjacent vertices in the flow contention graph should not be scheduled to transmit at the same time. Despite the differences of backoff algorithms and information exchange among these schemes, the underlying channel access scheme remains largely the basic sender-initiated collision avoidance handshake, which can be less effective than a receiver-initiated scheme when a receiver has better knowledge of the contention around itself than the sender.

Based on this key observation, in our earlier work [22], we proposed a hybrid channel access scheme that combines both sender-initiated and receiver-initiated collision avoidance handshake to address the fairness problem. The attractiveness of this approach is that it is compatible with the IEEE 802.11 framework and involves only some additional queue management and book-keeping work. However, this recent work has shown that, despite its simplicity, it is not very effective for TCP-based flows and that more information exchange among nodes is necessary to solve the fairness problem conclusively. This motivates us to further our work on a framework to address the fairness problem in a systematic way. In Section 2.1, we identify several key components that constitutes our fairness framework and explain the rationale for their necessity. In Section 2.2, we propose new algorithms to realize the fairness framework. The resulting scheme, which we simply call topology aware fair access (TAFA) is evaluated in Section 2.3 through computer simulations. The performance of

TAFa is compared with that of the original IEEE 802.11 MAC protocol and the hybrid channel access scheme proposed in [22] for both UDP- and TCP-based traffic. Simulation results show that TAFa can solve the fairness problem in UDP-based applications with negligible degradation in throughput. It can also solve the notorious problem of starvation of flows in TCP-based applications, despite some moderate degradation in throughput. Hence, TAFa shows a much better overall tradeoff between throughput and fairness than the other schemes investigated.

2.1 The Fairness Framework

In this section, we describe a framework for achieving better fairness consisting of four key components:

- Exchange of flow information among nodes;
- Adaptive backoff algorithm that is as stable as binary exponential backoff (BEB) but does not have the inherent deficiency of aggravating the fairness problem;
- Switching sender-initiated and receiver-initiated scheme as appropriate;
- Dealing with two-way flows.

The need for the exchange and maintenance of flow contention information can be illustrated by a simple example with the network configuration 4-8 shown in Figure 1.11. In Figure 1.11, a dashed line means that two nodes can hear each other's transmissions and an arrow indicates an active flow between two nodes. Nodes without any line in-between are hidden from each other. For configuration 4-8, node 2 knows that both node 0 and node 3 are sending nodes. However, if node 2 does not explicitly tell both node 0 and node 3 about the existence of each other, the handshake between node 0 and node 1 will tend to dominate the channel, because node 3's transmissions will mostly collide with either node 0 or node 1's transmissions at node 2, and both node 0 and node 1 may incorrectly perceive that node 0 and node 1 are the only active nodes in the network. Even though they may receive node 2's packets sporadically and make some ad hoc adjustment, without a systematic way to obtain flow information, the fairness problem cannot be solved conclusively.

The second component of our framework is an adaptive backoff scheme which is mandatory because the existing binary exponential backoff can aggravate the fairness problem as shown extensively in the literature [–21, 1, 14]. Nodes should decide their channel access based on the information of competing flows gathered through the first component.

The third component of our framework is a hybrid channel access scheme that combines both sender-initiated and receiver-initiated collision handshake. This is largely due to the advantage of distributing the burden of initiating collision avoidance handshake between a pair of sending and receiving nodes

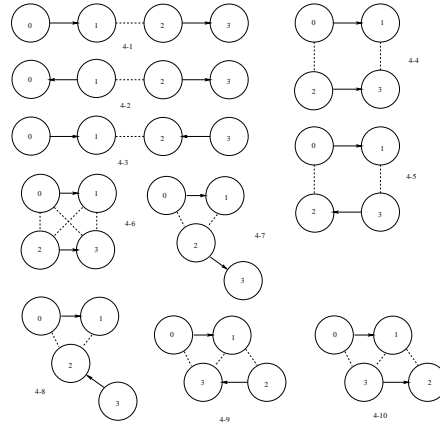


Figure 1.11. Network configurations with two competing flows

depending on the different degrees of contention they experience. For example, in the network configuration 4-1 shown in Figure 1.11, the flow from node 0 to node 1 will suffer severe throughput degradation if no proper action is taken, because RTS from node 2 can always be received by node 3 successfully while node 0's RTS collides with node 2's transmissions at node 1 most of the time. In this case, if the collision avoidance is initiated by node 1, which transmits CTS to node 0 directly, then the channel bandwidth will be shared between these two flows more evenly, because node 1 and node 2 are direct neighbors and it is easier for them to coordinate their access to the channel.

The fourth component of our framework is a key contribution of the framework and consists of dealing with two-way traffic in which there are one data flow and one acknowledgment flow between two nodes, as is the case in most TCP-based flows. In such cases, usually one node cannot continue sending data packets, unless it receives application level acknowledgment packets from the other node. Though viewed from a traditional MAC's perspective they are separate flows, the performance of these two flows is coupled and they should compete as a collective entity rather than do so separately. Fairness for such cases is only touched upon in [22] and has not been addressed adequately in the literature, because most of the performance evaluation of fair MAC schemes so far has been done with constant bit rate (CBR) like traffic. The information about whether a flow is one-way or has a reverse flow can be conveyed from the application down to the MAC layer through some interface, which is not discussed here. We believe that such information and hence the required special processing are necessary to achieve the desired fairness goal.

2.2 Topology-Aware Fair Access

The topology aware fair access (TAFA) scheme is a realization of the fairness framework described previously, and consists of four parts corresponding to the four components in the framework.

Exchange and Maintenance of Flow Information. Each node maintains a flow table and each entry in the table contains the following information about a flow: *source address*, *destination address*, *service tag*, *direct flag* and *position flag*.

The service tag is used to measure how much channel resource the flow has received. Though there can be several ways to calculate the service tag, we use a simple one, which consists of the number of bytes that have been sent by the sender and acknowledged by the receiver. The service tag is updated by the sender when it receives an acknowledgment from the receiver and updated information is propagated to other nodes through subsequent packet transmissions.

The direct flag is used to indicate whether the flow is known directly through listening to the channel or indirectly through flow advertisements from other nodes. For example, in the network configuration 4-8 shown in Figure 1.11, node 3 cannot know the flow from node 0 to node 1 directly and has to rely on node 2 to advertise that flow to it. In this case, the flow from node 0 to node 1 is recorded as indirect in node 3's flow table and node 3 does not advertise the indirect flow.

The position flag is used to indicate whether a flow is original, a derivative, or not applicable to either case. This flag is used to handle two-way traffic. For example, in some TCP-based applications, one end of the connection cannot continue sending packets, unless it receives a TCP acknowledgment from the other end. The MAC protocol cannot just treat the data flow and the acknowledgment flow as separate flows. Due to the asymmetry of most connections, i.e., a data flow usually generates much more traffic than the corresponding acknowledgment flow, trying to equate the channel utilization for both flows would lead to throughput degradation. So it is important to use the position flag to indicate whether the flow is *original* (data flow) or *derivative* (acknowledgment flow) and the service tag of a derivative flow should be adjusted according to that of the corresponding original flow.

In this scheme, an RTS or a CTS only carry the information about the current flow (from the sender to the receiver) to reduce the fixed overhead that exists whether fairness is desired or not. Because the source and destination of a flow is self evident and a direct flag is not necessary, the extra information included in the RTS and CTS is just the service tag and the position flag of the flow. A receiver just copies the service tag in an RTS to its outgoing CTS, so that the

neighbors of the receiver can also know the service tag of the ongoing flow. On the other hand, data packets and ACKs carry extra information about other flows maintained by the node if necessary. The rationale for treating these control packets differently is that the size of an RTS and a CTS can be fixed and nodes can get the duration information of the subsequent handshake from the network allocation vector (NAV) embedded in all packets. Because data packets are of varying size, it is acceptable for them to carry a bit more information. An ACK should also carry some extra flow information, otherwise those nodes that are neighbors of the node sending the ACK will never get any information about the flows around the node if the node does not send any data packet.

Specifically, to reduce the overhead incurred in the flow information exchange, nodes advertise only one flow at a time in the data or ACK packets they transmit, and one flow is chosen from the node's flow table in a round-robin way. As stated earlier, they only advertise flows that they know directly through receiving transmissions from either the sender or the receiver of the flow, rather than through the advertisement by other nodes. This avoids building up all the flows' information in a node which is unnecessary because channel access should be a local decision based only on the information of flows competing directly to avoid the complexity of making global decisions which is not what MAC layer should consider. Besides, nodes can obtain the updates of neighbor flows more quickly because only such flows are advertised. Flow information advertised in data and ACK packets includes only the source address, destination address and service tag.

Through the advertisement of flows, a node comes to know the other flows that may be competing with itself, gathers neighborhood topology information naturally, and adjusts its channel access accordingly.

Flow Aware Backoff Algorithm. In this scheme, each node also maintains two flags: *MyFlow* and *OtherFlow*. When a node receives the acknowledgment for its data packet, it updates its service tag and sets *MyFlow* true. When a node receives updated and greater service tag for other flows, it sets *OtherFlow* true. These two flags are used for a node to decide its contention window (CW), which is the upper bound of the uniform distribution from which a backoff timer is generated.

Unlike other schemes that deviate significantly from the binary exponential backoff (BEB) used in the IEEE 802.11 MAC protocol, we adopt BEB's basic idea of quick contention resolution and robustness and the resulting backoff algorithm is shown in Figure 1.12 in pseudo-code. Lines from 1 through 7 deal with the case when the node is the sender the flow with the minimum service tag. If neither the flow nor any other flow progresses (lines 2–3), then it means that some other nodes may also perceive that they have the minimum flows and it is important for the node to double its contention window (CW) for

```

1: if (My flow has the min service tag in my flow table) {
2:   if (!MyFlow && !OtherFlow)
3:     Double contention window;
4:   else if (OtherFlow)
5:     Keep current contention window;
6:   else if (MyFlow)
7:     Reset contention window to minimum;
8: } else {
9:   if (!MyFlow && !OtherFlow)
10:    Double contention window;
11:  else if (OtherFlow && !MyFlow)
12:    Keep current contention window;
13:  else if (MyFlow && !OtherFlow)
14:    Double contention window;
15:  else if (MyFlow && OtherFlow)
16:    Reset contention window to minimum;
17: }
18: Clear MyFlow and OtherFlow.

```

Figure 1.12. The adaptive backoff algorithm

quick contention resolution. If any other flow progresses (lines 4–5), then the node should keep its current CW lest it may cause collisions by decreasing the CW and suffer unfairness by increasing the current CW, because it is already lagging behind other flows. If this flow has already made progress (lines 6–7), then it is safe to set its CW to the minimum value, because there is no perceived immediate contention from other flows. Lines from 9 through 17 deal with the case when the node does not have the minimum flow. If neither my flow nor other flow progresses (lines 9–10), it is important to double the CW for quick contention resolution. If only other flows make progress (lines 11–12), then it is adequate to keep the current CW, because the node does not require immediate access to the channel. However, if only my flow progresses (lines 13–14), then it means that the node is too aggressive in its channel access and should double its CW to yield the channel access to the other nodes that have minimum flows. If both my flow and other flow progress, then the node can reset the CW to the minimum value to avoid too much time spent in backoff. At last, in line 18, both *MyFlow* and *OtherFlow* are cleared and the backoff algorithm will be adapted again to any future change made to these two flags.

Topology-Aware Hybrid Collision Avoidance Handshake. As we have discussed, sometimes receiver-initiated collision avoidance can be more effective than sender-initiated and a combination of both is shown to yield quite satisfactory results when used to address the fairness problem [22].

To put our scheme in perspective, we give a brief review of the hybrid channel access scheme proposed in [22]. To maintain its compatibility with IEEE 802.11, the hybrid scheme does not introduce new types of control packets. Instead, a CTS packet is reused as the polling packet. Hence, the receiver-initiated collision avoidance handshake just includes a three-way CTS-data-ACK exchange between polling and polled nodes.

Nodes implementing the hybrid scheme alternate in two modes: Sender-initiated (SI) and receive-initiated (RI). Nodes by default stay in the SI mode and use the usual four-way RTS-CTS-data-ACK handshake of IEEE 802.11. When a node transmits its RTS and fails to get the CTS from the intended receiver for several times, it sets the RI flag in the header of subsequent packets it sends and invites the receiver to start a receiver-initiated handshake. After receiving such packets, the receiver will confirm the sender with the RI flag also set in its reply if the receiver also implements the hybrid scheme. Upon receiving this confirmation, the sender will not transmit RTS packets to the receiver further. Instead, it just waits for the receiver to initiate a CTS-data-ACK handshake to itself, thus avoiding aggravating the contention for the channel. At this time, both nodes are engaged in the RI mode. A sender renews its RI request by setting RI flag continuously in the packets it sends out and cancels its request by clearing the flag, for example, when it has no more packet for the receiver.

The criterion to trigger the receiver-initiated handshake in [22] is that a node sets the RI request flag in its packets after it has sent the same RTS packet for more than one half of the times allowed in the IEEE 802.11 MAC protocol and receives no response from the intended receiver. The problem with this approach is that the receiver can hardly get any RTS sometimes due to high contention around it and hence receiver-initiated handshake cannot be triggered. This phenomenon is especially conspicuous for a two-way TCP connection, which consists of one data flow and one acknowledgment flow, because a pair of nodes may take turns to grab the channel, while other less privileged nodes may defer their access to the channel further due to the flow control and congestion avoidance functions in TCP.

To address the above problem, we propose a topology-aware scheme to switching between sender-initiated and receiver-initiated handshake. The basic idea is to make nodes that are closer to the contention initiate the handshake. To facilitate the description of the algorithm, some notations are used as shown in Table 1.4. Two flows are called dependent if they need to take turns to proceed, like a data flow and an acknowledgment flow in most TCP-based flows. That is why the position flag is exchanged and recorded in a node's flow table.

Figure 1.13 shows the criteria to switch between sender-initiated and receiver-initiated handshake. Similar to the algorithm shown in Figure 1.12, lines from 1 through 7 deal with the case when the node is the sender of the flow with the minimum service tag. If there is any independent flow in this node's table

Table 1.4. Notations used in the hybrid scheme

V	The node applying the algorithm
f_m	The flow with the minimum service tag among all the flows in node V 's flow table
f_{mi}	The flow with the minimum service tag among all the flows in node V 's flow table that are not dependent on flows originating from node V
$S(f)$	Sender of flow f
$R(f)$	Receiver of flow f
$N(V)$	Node V 's neighbors

(lines 2–6), then the node needs to differentiate between two cases. If either the sender or the receiver of the independent flow which has the minimum service tag is this node's neighbor (line 3), then the usual sender-initiated handshake is used (line 4). Otherwise, it is possible that the receiver of the node is closer to either the sender or the receiver of that independent flow and it is more appropriate for the node to ask its receiver to use receiver-initiated handshake (line 6). In this way, the node and its receiver may compete for the channel more effectively. If the node does not have the minimum flow (lines 8–13), it should find the minimum flow in its flow table first. If the node is the receiver of the minimum flow or either the sender or the receiver of the minimum flow is its neighbor, then it just stays in the SI mode (lines 9–11). Otherwise, it means that the receiver of its flow may be closer to the nodes having the minimum flow, and then the node asks its receiver to enter the RI mode (line 12) with the hope that its receiver may compete for the channel more effectively than itself.

Dealing with Two-Way Flows. Two-way flows require special processing as discussed before. We describe some necessary changes to the algorithms discussed in the previous subsections.

For an original flow and a derivative flow to compete for the channel effectively, the key idea is that the service tags for these flows in the participating nodes' flow tables should have correct relationship, i.e., if $T(f_1) \leq T(f_2)$ in one node's flow table, then it should be the same in the other node's flow table, so that nodes can make correct decisions in the backoff algorithm and the switch between sender-initiated and receiver-initiated handshake. It does not matter even if there are some discrepancies about the service tags of these flows maintained individually by each node.

```

1:  if ( $S(f_m)$  is  $V$ ) { // Our flow has the min service tag
2:      if ( $\exists f_{mi}$ ) { // If an independent flow is found;
3:          if ( $S(f_{mi}) \in N(V)$  or  $R(f_{mi}) \in N(V)$ )
4:              Sender-initiated;
5:          else
6:              Receiver-initiated;
7:          } else Sender-initiated;
8:  } else { // Some other flow has the min service tag
9:      if ( $V$  is  $R(f_m)$  or  $S(f_m) \in N(V)$ 
10:         or  $R(f_m) \in N(V)$ )
11:         Sender-initiated;
12:      else Receiver-initiated;
13:  }

```

Figure 1.13. The criteria to choose sender-initiated or receiver-initiated handshake

In dealing with two-way flows, it is important to differentiate between original and derivative flows: The original flow is the one from the node that initiates the connection to the other node that acknowledges the connection. Then the required special processing can be summarized in two rules.

Rule 1: When a node that initiates the original flow receives a packet from the corresponding derivative flow, it sets the service tag for the derivative flow (maintained in its flow table) to be the service tag of the original flow plus the size of the acknowledged data packet measured in bytes. It does not change the *OtherFlow* flag because in fact the derivative flow is not an independent flow.

Rule 2: When a node that is the sender of a derivative flow receives a packet from the corresponding original flow, it updates the service tags for both flows in its table as follows. Let f_o denote the received service tag of the original flow and f_d the current service tag of the derivative flow in its table. Then the new service tags for the original flow (f_{no}) and the derivative flow (f_{nd}) are: $f_{nd} = f_o$ and $f_{no} = f_o + f_d$. In this case, the node does not change *MyFlow* flag because the node itself is in effect not making any real progress.

Figure 1.14 shows the algorithm when the above two rules are applied.

How to apply these rules are better illustrated by the example shown in Table 1.5. In this example, the packet from the original flow (0→1) has a size of 100 bytes, and the packet from the derivative flow (1→0) has a size of 4 bytes. *My* and *Other* are the short names for the *MyFlow* and *OtherFlow* flags. It is clear that these two rules make sure that the service tags of these two flows have the correct relationship in either node's table even if they are not up-to-date.

```

1: if (My flow is original) {
2:   if (Receive a data packet  $T_d$  from the derivative flow)
3:      $f_{nd} = f_o + T_d$ ;
4: } else if (My flow is derivative) {
5:   if (Receive a data packet from the original flow)
6:      $f_{nd} = f_o$ ;  $f_{no} = f_o + f_d$ ;
7: }

```

Figure 1.14. Special tag processing for two-way flows

Table 1.5. An example of two-way flow processing

Time	Event	Node 0				Node 1			
		0→1	1→0	My	Other	0→1	1→0	My	Other
t_0	initialization	0	0	-	-	0	0	-	-
t_1	0 sends data, 1 acks	100	0	1	-	0	0	-	-
t_2	1 sends data, 0 acks	100	104	1	-	0	4	1	-
t_3	0 sends data	100	104	1	-	100	4	1	-
t_4	1 acks	200	104	1	-	104	100	1	-

2.3 Simulation Results

In our simulations, we focus on how two competing flows share the available channel resource in a few simple network configurations. These configurations are shown in Figure 1.11. Despite the simpleness of these configurations, it is interesting to note that the fair schemes [–1, 14, 15, 22] proposed so far have not addressed all the fairness problems in these network configurations when flows are either UDP- or TCP-based.

We use GloMoSim 2.0 [12] as the network simulator and our implementation of the new scheme (TAFA) is based on the IEEE 802.11 MAC protocol. Below are some details of the implementation. For RTS/CTS, we add three fields: service tag (4 bytes), position flag (2 byte) of current flow and receiver-initiated (RI) flag (2 byte). Though 1 byte should be enough for any of these flags, we choose larger size to allow easy extensions in the future if any. For data and ACK, in addition to the above three fields, they also include an advertisement about a flow from its flow table which includes three fields: source address (4 bytes), destination address (4 bytes) and service tag (4 bytes). In our implementation, a node indicates explicitly its originality in the RTS/data packets it sends out if applicable. All these constitute the fixed packet overhead in using the new scheme.

Table 1.6. IEEE 802.11 and TAFa specific configuration parameters

	<i>RTS</i>	<i>CTS</i>	<i>data header</i>	<i>ACK</i>
802.11	20-byte	14-byte	28-byte	14-byte
TAFa	28-byte	22-byte	48-byte	34-byte

For IEEE 802.11, direct sequence spread spectrum (DSSS) parameters are used throughout the simulations. Most of the parameters remain the same as shown in Table 1.1 and protocol specific configuration parameters are shown in Table 1.6.

We investigate the performance of the IEEE 802.11 MAC protocol, the hybrid channel access scheme (for simplicity, it is simply called *Hybrid* thereafter) and the TAFa scheme under both UDP- and TCP-based traffic. In the first set of the simulation experiments, there are two competing UDP-based flows. For each flow, one node keeps sending data packets to the other at a constant bit rate, such that the sending queue is always non-empty. UDP is the underlying transport layer, thus no acknowledgment packet is sent back to the initiating node. We ran each configuration five times with different seed numbers and with a duration of 30 seconds. If the standard deviation of throughput is within 10% of the mean throughput, we show mean values only. Otherwise, we show both mean and standard deviation of the throughput. Table 1.7 shows the configurations when the IEEE 802.11 MAC protocol has fairness problem or there is some difference among these schemes. The “-” sign in the rows for the hybrid scheme indicates that receiver-initiated handshake is not triggered at all. It can be seen that in some configurations such as 4-3 and 4-9 when the existing IEEE 802.11 MAC protocol works well, both the hybrid scheme and TAFa are unnecessary. Still, throughput degradation in TAFa is negligible. On the other hand, in configurations such as 4-1 and 4-8 where serious fairness problems occur in 802.11, TAFa shows superior performance to the other two schemes.

In the second set of simulation experiments, there are two competing TCP-based flows. We use the FTP/Generic application provided in GloMoSim, in which a client simply sends data packets to a server without the server sending any control information back to the client other than the acknowledgment packets required by TCP. Whenever a packet indicates success of delivery by the transport layer (TCP), the client sends the next data packet. It should be noted that the acknowledgment packet from TCP is still regarded as a normal data packet from the view of traditional MAC layer, which does not provide special processing for two-way flows. However, in TAFa, the data flow and the acknowledgment flow are regarded as the original flow and derivative flow,

Table 1.7. Throughput comparison for the IEEE 802.11, the hybrid scheme and TAFa – two CBR flows (throughput measured in kbps)

Config #	Scheme	Flow #	Throughput	Flow #	Throughput	Aggregate
4-1	802.11	0 → 1	83.4	2 → 3	1500	1580
	Hybrid	0 → 1	369	2 → 3	1230	1600
	TAFa	0 → 1	771	2 → 3	778	1550
4-2	802.11	1 → 0	820	2 → 3	814	1630
	Hybrid	1 → 0	-	2 → 3	-	-
	TAFa	0 → 1	769	3 → 2	769	1540
4-3	802.11	0 → 1	688	3 → 2	709	1400
	Hybrid	0 → 1	665	3 → 2	643	1310
	TAFa	0 → 1	683	3 → 2	656	1340
4-7	802.11	0 → 1	783	2 → 3	824	1610
	Hybrid	0 → 1	-	2 → 3	-	-
	TAFa	0 → 1	764	2 → 3	764	1530
4-8	802.11	0 → 1	1550	3 → 2	28	1580
	Hybrid	0 → 1	1280	3 → 2	319	1600
	TAFa	0 → 1	773	3 → 2	805	1580
4-9	802.11	0 → 1	734	2 → 3	809	1540
	Hybrid	0 → 1	815	2 → 3	742	1560
	TAFa	0 → 1	681	2 → 3	676	1360

respectively, and special processing is invoked as discussed in Section 2.2.0. Simulation results are shown in Table 1.8 for only the configurations when the IEEE 802.11 MAC protocol has fairness problems.

It is clear from Table 1.8 that the fairness problem is much more severe for two competing TCP-based flows than for the case of UDP-based flows if no special processing is in place. For example, in some cases, such as configurations 4-1, 4-7 and 4-8, one FTP flow is denied access to the shared channel for most of the time. The hybrid scheme, due to the lack of flow contention information, cannot trigger the desired receiver-initiated collision avoidance handshake, hence it is of no avail. On the other hand, TAFa achieves much better fairness though at a cost of degraded throughput. This is a much desired tradeoff because it avoids the starvation of some flows and hence channel bandwidth is more evenly distributed among participating nodes. For configuration 4-3, please note the high variation of the throughput for these two flows in the case of IEEE 802.11 which shows that one flow monopolizes the channel for a long time and then gives it away to the other flow. Both the hybrid scheme and TAFa help to solve the problem. For other configurations, TAFa suffers some degradation in throughput. However, the overall performance of TAFa shows a much better tradeoff between throughput and fairness among the three schemes we

Table 1.8. Throughput comparison for the IEEE 802.11, the hybrid scheme and TAFa – two FTP flows (throughput measured in kbps)

Config #	Scheme	Flow #	Throughput	Flow #	Throughput	Aggregate
4-1	802.11	0 → 1	0	2 → 3	926	929
	Hybrid	0 → 1	-	2 → 3	-	-
	TAFa	0 → 1	249	2 → 3	423	672
4-2	802.11	1 → 0	488±103	2 → 3	453±102	942
	Hybrid	0 → 1	439±99	3 → 2	502±98	940
	TAFa	0 → 1	383	3 → 2	390	773
4-3	802.11	0 → 1	530±432	3 → 2	392±438	922
	Hybrid	0 → 1	397±71	3 → 2	455±78	852
	TAFa	0 → 1	272	3 → 2	363	635
4-7	802.11	0 → 1	928	2 → 3	0	930
	Hybrid	0 → 1	-	2 → 3	-	-
	TAFa	0 → 1	443	2 → 3	332	775
4-8	802.11	0 → 1	929	3 → 2	0	930
	Hybrid	0 → 1	-	3 → 2	-	-
	TAFa	0 → 1	409	3 → 2	209	617
4-10	802.11	0 → 1	376	3 → 2	526	902
	Hybrid	0 → 1	-	3 → 2	-	-
	TAFa	0 → 1	335	3 → 2	438	773

investigate. We expect that even better algorithms than TAFa can be designed in the future following our fairness framework.

3. Conclusion

In this chapter, we have presented our work on throughput and fairness of collision avoidance protocols in ad hoc networks.

In the first part of our work, we use a simple model to derive the saturation throughput of MAC protocols based on an RTS-CTS-data-ACK handshake in multi-hop networks. The results show that these protocols outperform CSMA protocols, even when the overhead of RTS/CTS exchange is rather high, thus showing the importance of correct collision avoidance in random access protocols. More importantly, it is shown that the overall performance of the sender-initiated collision avoidance scheme degrades rather rapidly when the number of competing nodes allowed within a region increases, in contrast to the case of fully-connected networks and networks with limited hidden terminals reported in the literature [11, 10, 13], where throughput remains almost the same for a large number of nodes. The significance of the analysis is that the scalability problem of contention-based collision-avoidance MAC protocols looms much earlier than people might expect. Simulation experiments with the IEEE 802.11 MAC protocol validate these observations and show that the IEEE 802.11 MAC

protocol can suffer severe degradation in throughput due to its inability to avoid collisions between data packets and other packets even when the number of competing nodes in a region is small. However, when the number of competing nodes in a region increases, the performance gap is smaller as perfect collision avoidance protocols also begins to suffer from exceedingly long waiting time.

In the second part of our work, we propose a framework to address the fairness problem in ad hoc networks systematically. The framework includes four key components: Exchange of flow contention information, adaptive and stable backoff algorithm, hybrid collision avoidance handshake, and special processing for two-way flows. We proposed some specific algorithms to realize the framework and the resulting scheme, called topology aware fair access (TAFa), was evaluated through computer simulations against the IEEE 802.11 MAC protocol and a hybrid channel scheme proposed earlier in the literature. It was shown that TAFa can solve the fairness problem in UDP-based applications with negligible degradation in throughput. TAFa is also quite promising for TCP-based applications, which have not been investigated at length in the past. Though TAFa suffers some throughput degradation, it solves the notorious problem of starvation of TCP flows, thus showing a much better overall tradeoff between throughput and fairness than the other schemes.

Given that the fairness framework is tailored to ad hoc networks and is general enough to accommodate new algorithms, it will be interesting to investigate new adaptive backoff algorithm and new criteria to switch between sender-initiated and receiver-initiated collision avoidance to achieve better throughput and fairness tradeoffs in future work.

Notes

1. The curves for $N = 3$ with different values of α concentrates on the upper part of these figures while the ones for $N = 10$ on the lower part.
2. Here we refer to those nodes that have at least one common neighbor with a node but are not direct neighbors of the node as the node's two-hop neighbors.

References

- [1] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LANs," in *Proc. of ACM SIGCOMM '94*, 1994.
- [2] IEEE Computer Society LAN MAN Standards Committee, ed., *IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE Std 802.11-1997, The Institute of Electrical and Electronics Engineers, New York, 1997.
- [3] J. J. Garcia-Luna-Aceves and C. L. Fullmer, "Floor Acquisition Multiple Access (FAMA) in Single-channel Wireless Networks," *ACM/Baltzer*

Mobile Networks and Applications, vol. 4, no. 3, pp. 157–174, 1999.

- [4] F. Talucci and M. Gerla, “MACA-BI (MACA by Invitation): A Receiver Oriented Access Protocol for Wireless Multihop Networks,” in *Proc. of PIMRC '97*, 1997.
- [5] J. J. Garcia-Luna-Aceves and A. Tzamaloukas, “Receiver-initiated Collision Avoidance in Wireless Networks,” *ACM Wireless Networks*, vol. 8, pp. 249–263, 2002.
- [6] F. A. Tobagi and L. Kleinrock, “Packet Switching in Radio Channels: Part II - the Hidden Terminal Problem in Carrier Sense Multiple-access Modes and the Busy-tone Solution,” *IEEE Trans. on Communications*, vol. 23, no. 12, pp. 1417–1433, 1975.
- [7] Y. Wang and J. J. Garcia-Luna-Aceves, “Performance of Collision Avoidance Protocols in Single-Channel Ad Hoc Networks,” in *Proc. of IEEE Intl. Conf. on Network Protocols (ICNP '02)*, (Paris, France), Nov. 2002.
- [8] H. Takagi and L. Kleinrock, “Optimal Transmission Range for Randomly Distributed Packet Radio Terminals,” *IEEE Transactions on Communications*, vol. 32, no. 3, pp. 246–57, 1984.
- [9] L. Wu and P. Varshney, “Performance Analysis of CSMA and BTMA Protocols in Multihop Networks (I). Single Channel Case,” *Information Sciences, Elsevier Sciences Inc.*, vol. 120, pp. 159–77, 1999.
- [10] F. Cali, M. Conti, and E. Gregori, “Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit,” *IEEE/ACM Transactions on Networking*, vol. 8, pp. 785–799, Dec. 2000.
- [11] G. Bianchi, “Performance Analysis of the IEEE 802.11 Distributed Coordination Function,” *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 535–547, Mar. 2000.
- [12] X. Zeng, R. Bagrodia, and M. Gerla, “GloMoSim: a Library for Parallel Simulation of Large-scale Wireless Networks,” in *Proc. of the 12th Workshop on Parallel and Distributed Simulations*, May 1998.
- [13] F. Cali, M. Conti, and E. Gregori, “IEEE 802.11 Protocol: Design and Performance Evaluation of an Adaptive Backoff Mechanism,” *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 1774–1786, Sept. 2000.
- [14] T. Ozugur, M. Naghshineh, P. Kermani, C. M. Olsen, B. Rezvani, and J. A. Copeland, “Balanced Media Access Methods for Wireless Networks,” in *Proc. of ACM/IEEE MOBICOM '98*, pp. 21–32, Oct. 1998.
- [15] B. Bensaou, Y. Wang, and C. C. Ko, “Fair Medium Access in 802.11 Based Wireless Ad-Hoc Networks,” in *IEEE/ACM Intl. Workshop on Mobile Ad Hoc Networking and Computing (MobiHoc '00)*, (Boston, MA, U.S.A.), Aug. 2000.

- [16] X. Huang and B. Bensaou, "On Max-min Fairness and Scheduling in Wireless Ad-Hoc Networks: Analytical Framework and Implementation," in *ACM MobiHoc '01*, (Long Beach, CA, U.S.A.), Oct. 2001.
- [17] N. H. Vaidya, P. Bahl, and S. Gupta, "Distributed Fair Scheduling in a Wireless LAN," in *ACM Mobicom 2000*, (Boston, MA, USA), Aug. 2000.
- [18] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan, "Achieving MAC Layer Fairness in Wireless Packet Networks," in *ACM Mobicom 2000*, (Boston, MA, USA), Aug. 2000.
- [19] H. Luo, S. Lu, and V. Bharghavan, "A New Model for Packet Scheduling in Multihop Wireless Networks," in *ACM Mobicom 2000*, (Boston, MA, USA), Aug. 2000.
- [20] H. Luo and S. Lu, "A Topology-Independent Fair Queueing Model in Ad Hoc Wireless Networks," in *IEEE ICNP 2000*, (Osaka, Japan), Nov. 2000.
- [21] H. Luo, P. Medvedev, J. Cheng, and S. Lu, "A Self-Coordinating Approach to Distributed Fair Queueing in Ad Hoc Wireless Networks," in *IEEE INFOCOM 2001*, Apr. 2001.
- [22] Y. Wang and J. J. Garcia-Luna-Aceves, "A New Hybrid Channel Access Scheme for Ad Hoc Networks," *ACM Wireless Networks Journal, Special Issue on Ad Hoc Networking*, vol. 10, no. 4, 2004.