FAIRNESS AND SCHEDULING IN AD HOC NETWORKS

JACOB ESHET AND BEN LIANG*

Abstract. Reliable medium access in ad hoc networks is a necessary step in the development of this promising technology. However, the Medium Access Control layer in a distributed network is difficult to manage. The shared channel and limited information available at each node make collisions and unfairness a common occurrence that has to be dealt with. The ideal protocol will provide the maximum amount of throughput for each flow in the network. This chapter will present some of the better known MAC protocols, fairness mechanisms and scheduling algorithms. The advantages and disadvantages of each will be highlighted, and simulation results will be used to provide a fair comparison.

1. Introduction. Ad hoc networking is currently a widely researched topic. The prospect of having a multitude of wireless devices communicating with each other without the need for fixed infrastructure is very exciting. Ad hoc networks may very well revolutionize the way we live in the not so distant future. Some critics question the feasibility of a full scale general purpose ad hoc network. Still, when a communication infrastructure does not exist or has been damaged, an ad hoc network may be the only alternative. Possible uses of ad hoc networks include:

Military/Natural Disasters Dangerous and/or complicated operations where a communications infrastructure is not available.

Sensor Networks Limited and well defined applications where mobility is not an issue.

Small Scale Temporary Deployment such as conferences, where constructing a communications infrastructure is not financially viable.

Even for limited applications, some of which are described above, the design of an ad aoc network is not trivial. The distributed wireless 'ad hoc environment' is harder to manage than a wired or centralized wireless environment. Additional factors that have to be considered in ad hoc networks include:

Local Congestion Congestion is dependent on topology and may vary widely over short distances.

Shared Medium The medium is shared by all nodes. Nodes can interfere with each other without being aware of each other's existence.

Mobility A node's operation has to dynamically adjust to a continuously changing topology.

Global Information Information on the entire network, which is needed by many algorithms for synchronization and other purposes is not available.

These factors have led to considerable research and new protocols devoted to ad hoc networks, as the existing protocols were no longer suitable.

MAC Layer. The Medium Access Control (MAC) layer is the lower sub layer of the data link layer in the OSI 7 Layer model (see figure 1.1). This layer includes mechanisms and protocols that determine when and how a node accesses the channel. The ideal protocol will permit nodes to access the channel in such a way that

*Edward S. Rogers Department of Electrical and Computer Engineering, University of Toronto, 10 King's College Road, Toronto, ON, Canada, M5S 3G4, e-mail: {eshet,liang}@comm.utoronto.ca

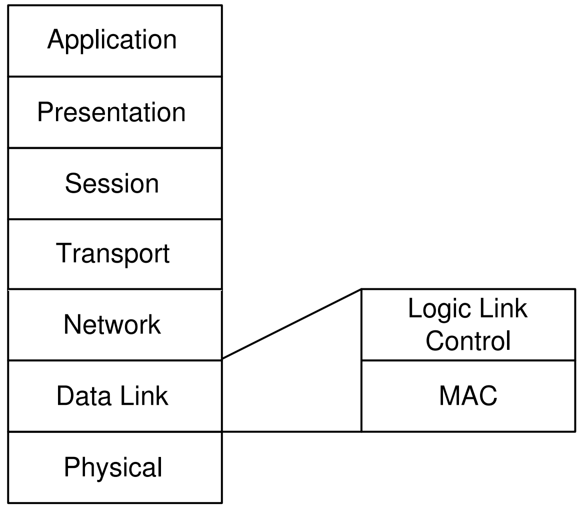

FIG. 1.1. *The OSI 7 Layer Model*

throughput will be maximized and fairly divided between flows. This goal is difficult to achieve for several reasons. First, it is not well defined, since fairness can mean different things to different people. Second, throughput and fairness are often a trade off. Usually, maximum aggregate throughput can be achieved by letting some transmitters send constantly and not letting the remaining transmitters send at all. This is clearly an unacceptable solution. This is why, a lower aggregate throughput must be accepted in favour of better fairness. The reduction in throughput that must be tolerated depends on how fairness is defined. Furthermore, even if the ideal amount of throughput and fairness can be ascertained, achieving this throughput and fairness in a distributed environment is not a simple task. Typically, some amount of overhead information has to be exchanged between nodes to achieve better fairness, and this results in a further reduction in throughput. In the next few sections several protocols will be examined in terms of how they define fairness, and how well they do in achieving their goal.

The absence of a centralized coordinator in ad hoc networks, forces terminals to compete against each other for the channel. Since data is sent from the transmitter to the receiver, the transmitter should ensure that the receiver can receive its transmission without interference, i.e. that there is no other (interfering) signal that the receiver is receiving simultaneously.[1] However, the transmitter is often not aware of such interfering signals because it is too far away from the interfering transmitter. The resulting contention will cause severe degradation in throughput, and often result in an unfair allocation of throughput. Much of the degradation in throughput and fairness is attributed to the well known *hidden terminal problem.* This problem will be illustrated with the help of figure 1.2. Consider the case when terminal A is transmitting to terminal B. Terminal C is out of range of terminal A and is therefore not aware of the transmission taking place. Terminal C might decide to transmit at this time, thus creating interference at terminal B. Both transmissions will continue and waste precious bandwidth. Solving the hidden terminal problem is very important

[1] Assuming that nodes are not equipped with multi-packet reception technology.

FIG. 1.2. *The Hidden and Exposed Terminal Problems*

to the performance of a MAC protocol, since this troublesome scenario occurs often. Another related and often ignored problem is known as the *exposed terminal problem.* Refer back to figure 2, and consider the case where terminal B is transmitting to terminal A. Theoretically, there is no reason why terminal C should not transmit to any terminal other than B at this time. This would increase the throughput of the network. In practice, however, terminal C will often (depending on the MAC protocol) detect the transmission from terminal B and defer its transmission. Protocols that solve the exposed terminal problem achieve better throughput.

In order to resolve contentions, a mechanism involving random access must be used. To see this, note that a terminal becomes aware of a contention when an expected response such as an acknowledgement does not arrive. If the node retransmitted the original packet immediately after becoming aware of the contention, this would most likely result in another collision, since the interferer would also retransmit immediately and additional transmissions may begin at any time making congestion even worse. This situation is solved by most protocols by using a *backoff* mechanism. When detecting a contention a terminal backs off for a random period of time. The terminal that picks the shorter time will usually capture the channel first. The random period of time is derived using an algorithm that considers some network parameters and some random component. It is the responsibility of the MAC protocol to resolve contentions in a manner that will provide maximum throughput and divide the throughput fairly among terminals. In section 2, several MAC protocols will be presented. An emphasis will be placed on how each protocol deals with contentions and how it performs in terms of fairness. Section 3 will present additional fairness algorithms that can be integrated into other MAC protocols. Section 4 will introduce the concept of scheduling and will describe several scheduling algorithms. Section 5 will summarize and compare all of the protocols discussed in this chapter, and section 6 will conclude the chapter.

2. MAC Protocols. This section describes some of the better known MAC protocols. Each of these protocols made a significant contribution to MAC layer protocol development in ad hoc networks.

2.1. Aloha. The Aloha network was developed in the 1970s at the University of Hawaii ([2]). Its purpose was to connect terminals located on different islands. The Aloha protocol follows a very simple algorithm. A terminal sends out a packet (DATA) as soon as it becomes available. The receiver replies with an acknowledgement (ACK). If an acknowledgement is not received, the terminal assumes a collision occurred and retransmits the packet after an exponentially distributed random period of time. The original network was fully connected, which simplifies the analysis greatly.

For example, in the analysis presented in [15], the length of a data packet is assumed to be a constant X seconds. If the arrival rate of new packets into the network is then denotes by S packets/X seconds and the total arrival rate (new packets and retransmitted packets, also known as load) is denoted by G packets/X

seconds then

$$S = GP[nocollision] \; .$$

The probability of no collision is the probability that 0 packets will be sent in $2X$ seconds. To see this refer to figure 2.1 and note that any packet sent between $t_0 - X$ and $t_0 + X$ will interfere with a packet sent at time t_0. If the arrival process is then

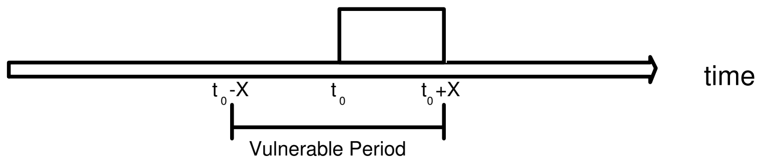

FIG. 2.1. *Vulnerable Period in Aloha*

assumed to be Poisson with parameter $2G$ ($2G$ arrivals in $2X$ seconds), then

$$P[k \; transmissions \; in \; 2X \; seconds] = \frac{(2G)^k}{k} e^{-2G} \; .$$

Finally, we have,

$$\begin{aligned} S &= GP[no \; collision] \\ &= GP[0 \; transmissions \; in \; 2X \; seconds] \\ &= Ge^{-2G} \; . \end{aligned}$$

A plot of throughput vs. load for Aloha is shown in figure 2.2. From this plot it is clear that the maximum throughput (S) of this system is 18.4% when the applied load (G) is 50% of channel capacity.

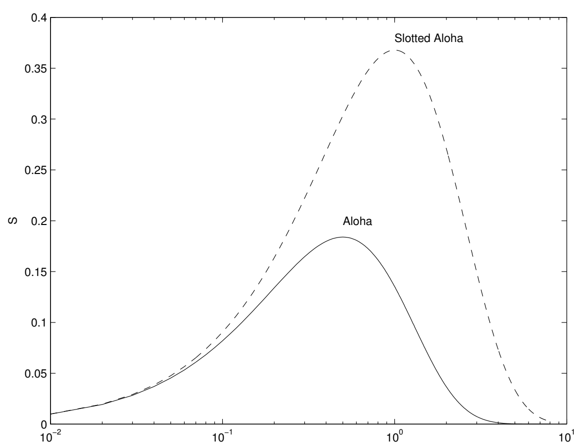

FIG. 2.2. *Aloha and Slotted Aloha: Throughput vs. Load*

2.1.1. Slotted Aloha. The extremely low throughput of Aloha can be improved by slotting time so that a packet can only be sent at the beginning of a slot. This modified scheme is called Slotted Aloha. If the slot length is the length of a packet (X seconds), an analysis similar to the analysis of Aloha can proceed as follows:

$$S = GP[\textit{no collision}]$$

as before. However, the vulnerable period is only X seconds. To see this refer to figure 2.3 and note that if a packet is sent at time t_0, there will be no collisions if and only if no other packet arrives at any node between times $t_0 - X$ and t_0. The shorter vulnerable period increases the probability of no collisions, which is the reason for Slotted Aloha's improved performance. The arrival process can similarly be assumed

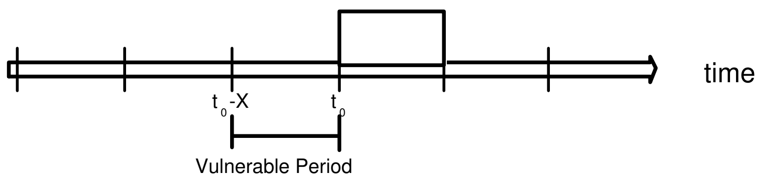

FIG. 2.3. *Vulnerable Period in Slotted Aloha*

to be Poisson with parameter G (G arrivals in X seconds), and

$$P[k \textit{ transmissions in } X \textit{ seconds}] = \frac{(G)^k}{k} e^{-G} .$$

Finally,

$$\begin{aligned} S &= GP[\textit{no collision}] \\ &= GP[0 \textit{ transmissions in } X \textit{ seconds}] \\ &= Ge^{-G} . \end{aligned}$$

A plot of throughput vs. load for Slotted Aloha is shown in figure 2.2 together with the Aloha plot. The plot shows that in this scheme a maximum throughput of 36.8% is obtained when the applied load (G) is equal to the channel capacity.

The Aloha and Slotted Aloha analyses yield very low maximum throughput values. These values are not guaranteed and can only be achieved by adjusting the applied load to 1/2 channel capacity and full channel capacity for Aloha and Slotted Aloha respectively. This can only be done by actively controlling new traffic transmissions and modifying the backoff period. However, overall load is a parameter that is not available to individual terminals in an ad hoc network, which makes this impossible to do in practice. Furthermore, in a non fully connected network, the concept of applied load has to be adjusted to account for varying loads across the network, complicating the matter even more.

Despite the low throughput, however, it is clear that both Aloha and Slotted Aloha achieve perfect fairness. This is because every terminal is equally likely to win a contention. In a non fully connected network, nodes that have more neighbours will receive less throughput. This is in line with the intuitive meaning of fairness.

2.2. CSMA. Carrier Sense Multiple Access (CSMA) is a mechanism that requires each terminal to be equipped with hardware that can sense whether the channel is in use at the physical layer. If the channel is busy, the terminal can avoid transmitting and prevent collisions. CSMA exists in several flavours. CSMA-CA is the most common MAC protocol mechanism in ad hoc networks. It often forms the basis for other MAC protocols.

2.2.1. *p*-persistent CSMA. In p-persistent CSMA, a terminal senses the channel until it is idle. When the channel is idle the terminal transmits with probability p, or backs off with probability $1-p$ for a random amount of time. If $p = 1$, terminals transmit immediately when the channel is idle. A single collision will then result in further congestion that will only increase with time. If $p = 0$, then when the channel becomes idle, a terminal always backs off and transmits when the backoff expires if the channel is still idle. Choosing a high p will increase congestion but will also increase channel utilization. Choosing a low p will reduce congestion, but the channel may become under utilized. Clearly the optimal value of p is dependent on the specific scenario. Topology and traffic arrival patterns must be taken into account. Even for a specific scenario, the optimal value of p will differ between nodes, and may even change over time. Finding the optimal value of p is the main obstacle of this mechanism. Some mechanisms that will be described more fully in section 3, focus on finding the optimal p.

2.2.2. CSMA-CA. Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) serves as the foundation for many recent MAC protocols. This mechanism borrows two control packets from MACA ([14]): RTS (Request To Send) and CTS (Clear To Send). Both of these packets are significantly shorter than the DATA packet and protect the DATA packet by signalling to neighbouring nodes that a transmission is about to take place. When terminal A wants to transmit data to terminal B, it waits for the channel to become idle. When the channel is idle A sends an RTS and waits for B to return a CTS. If terminal B does not return a CTS within a specified amount of time, terminal A assumes a collision has occurred and backs off. If terminal B returns a CTS, A proceeds to send the DATA packet. Once the DATA packet has been sent, terminal B will acknowledge the reception with an ACK packet. If an ACK packet is not received in time, terminal A assumes the DATA wasn't received, backs off, and retransmits the RTS at a later time.

The RTS and CTS packets include information on the remaining duration of the RTS-CTS-DATA-ACK exchange. Neighbouring nodes of the transmitter or receiver will ideally overhear either the RTS or the CTS. They then become aware that a transmission is taking place in their vicinity and how long the transmission will last. Since they are not allowed to transmit until this exchange is done, they set up a Network Allocation Vector (NAV) for this duration. While the NAV is in place, they will not begin a transmission or reply to another transmission. This is also known as virtual carrier sensing. The RTS and CTS packets solve the hidden terminal problem by informing the hidden transmitter of the transmission about to take place. However, it is possible that the RTS or CTS packets will collide with other control or data packets and will not be overheard by the hidden transmitter. In more complicated scenarios, RTS or CTS packets may not be overheard due to mobility or power saving. By virtually eliminating the hidden terminal problem, CSMA-CA significantly increases throughput, but the effects on fairness are dependent on its exact implementation. The most famous implementation of CSMA-CA is called DFWMAC and is described in the next section. Note that CSMA-CA does not solve the exposed terminal problem. This is because the transmitter must wait for an ACK from the receiver. If neighbours of the transmitter were to start transmitting before the ACK is received, the transmitter would not know if its DATA was received.

2.2.3. DFWMAC (802.11). Distributed Foundation Wireless Medium Access Control (DFWMAC) also known as 802.11 is a standard developed by IEEE for wireless LANS. This standard is used in all commercial wireless LAN cards that are 802.11b compatible. DFWMAC uses the CSMA-CA mechanism, but is discussed here separately because of its immense popularity in ad hoc networks simulations and research. The standard, which is fully described in [1], can function in a distributed mode known as DCF or a centralized mode known as PCF. Only the DCF mode will be discussed here. The standard includes details on the waiting periods between packet reception and transmission of a new packet known as Inter Frame Spacing (IFS). DIFS is the period of time the channel must be idle before a transmitter begins a new exchange. SIFS is the period of time the channel must be idle before responding to an ongoing exchange, for example sending a CTS in response to an RTS or sending DATA in response to a CTS. The standard sets the time of SIFS to be shorter than DIFS so that a terminal replying to an ongoing exchange will have priority over a terminal beginning a new exchange. Another important element of the DFWMAC implementation is its backoff mechanism, known as Binary Exponential Backoff (BEB). A terminal must backoff when a collision occurs. The backoff time is chosen randomly with uniform distribution between 0 and CW. When a collision occurs, CW is doubled up to a maximum value of CW_{max}. If the transmission is successful CW is reset to CW_{min}. The logic behind BEB is that if the network is congested, backoff window sizes should be increased and vice versa.

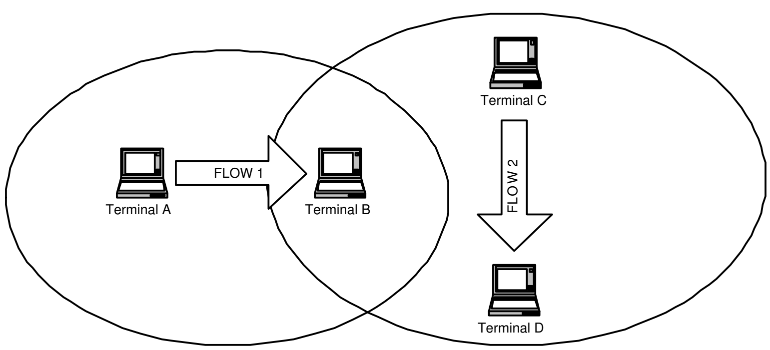

FIG. 2.4. *DFWMAC and the capture effect*

DFWMAC is notorious for its unfairness. Refer to figure 2.4 and consider the case when terminal A has packets to send to terminal B and terminal C has packets to send to terminal D. If terminal A captures the channel, terminal C becomes aware of this when it overhears the CTS from terminal B. It follows that terminal C sets a NAV and defers its transmission until terminal A's transmission is done. When terminal A is done, both terminals A and C have an equal chance of capturing the channel. On the other hand, if terminal C captures the channel, terminal A does not become aware of this since it is out of range of terminals C and D. Therefore, terminal A will send out an RTS. Terminal B cannot respond with a CTS since it is within the range of terminals C and D. Terminal A will assume a collision has occurred and will double its backoff window size. This will continue, as A sends more RTSs and keeps doubling its backoff window size. When terminal C's transmission is done, C will have a much better chance of capturing the channel again since its backoff window has the minimum size. In this scenario flow 2 will capture the channel for the majority of the time and flow 1 will be severely backlogged. Figure 2.5 shows simulation results for DFWMAC with this topology. The simulation was carried out for 50s. Note that flow 2 gets 99.2% of the throughput. This unfairness is particularly troublesome because the topology is very simple and may occur frequently in real life scenarios. Other simulations [16] show that a dense ad hoc network using the 802.11 MAC protocol has a throughput of only a few percent of channel capacity.

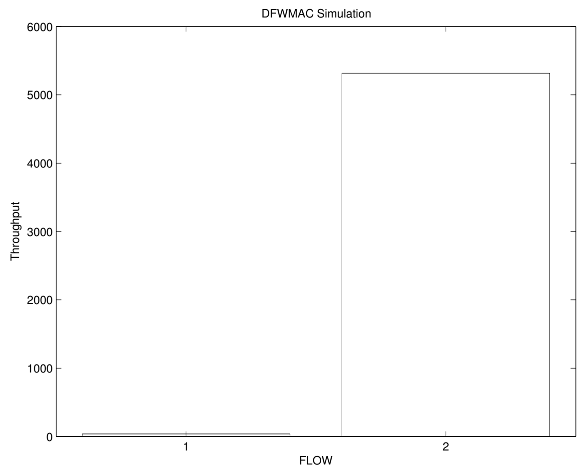

FIG. 2.5. *DFWMAC Simulation of the topology in figure 2.4.*

2.3. MACAW. MACAW, proposed by Bharghavan et al. in [5], is an improved version of Multiple Access Collision Avoidance (MACA) proposed by Karn in [14]. This MAC protocol focuses on the issue of fairness and is often given credit for being the first paper to point out the fairness problem. MACAW begins by considering the backoff algorithm and presents several significant innovations. First, to prevent one flow from capturing the channel while other flows have a large CW parameter, a new field is added to the header of each packet. This field contains the CW parameter of the transmitter. Terminals overhearing this packet copy this value into their own CW parameter ensuring all flows have an equal chance of capturing the channel in the next transmission. Second, to prevent large oscillations in the CW parameter, a Multiple Increase Linear Decrease (MILD) backoff algorithm is implemented. Upon successful transmission CW is decreased by one unit. If a collision occurs, $CW = 1.5 \times CW$. The paper also considers keeping a separate CW parameter for each flow. This allows the transmitter to keep separate congestion measurements for different areas of the network. MACAW also includes two new control packets, DS and RRTS. The DS packet is intended to replace carrier sensing hardware while achieving the same effect. The DS packet is sent by the transmitter after receiving the CTS and before sending the DATA packet. It signals to neighbouring transmitters that the RTS-CTS exchange was successful and data is about to be sent. If a neighbouring terminal does not hear a DS in time, it may assume the RTS-CTS exchange was not successful and might begin its own transmission. The RRTS packet is sent by a receiver that received an RTS but was not able to reply with a CTS because its NAV was in place. The RRTS invites the transmitter to resend the RTS.

While MACAW improves fairness considerably over DFWMAC, the addition of extra packets degrade throughput and increase the chance of control packet collisions. Some problems are also not solved by MACAW, for example the problem depicted in figure 2.4. The MILD backoff algorithm improves fairness slightly, but in the long run flow 1 will still capture the channel for the majority of the time. This is because information about the existence of flow 2 never reaches the transmitter of flow 1.

2.4. Busy Tones. Busy tones are narrow bandwidth out of band signals. Implementing busy tones requires more sophisticated hardware but the advantages provided by a busy tone cannot be achieved with in-band signaling. The original Busy Tone Multiple Access (BTMA) paper [22] proposed a protocol that uses a busy tone in a centralized network. The base station would send a busy tone whenever it detected a transmission from any terminal. Terminals hearing a busy tone would not begin to transmit, and thus would not interfere with each other.

2.4.1. RI-BTMA. The BTMA protocol was extended by Wu and Li for use in ad hoc networks in [23]. This extended protocol, named Receiver Initiated Busy Tone Multiple Access (RI-BTMA), used a receiver busy tone and time slotted operation. The transmitter may send a preamble to the receiver only at the beginning of a time slot. Upon receiving the preamble, the receiver emits the busy tone. Once the busy tone is heard by the transmitter, the transmitter sends the DATA packet. A transmitter may not send the preamble if a busy tone is detected.

The busy tone serves as an original and powerful solution to the hidden terminal problem. Unlike in CSMA-CA, where the CTS can be missed by a hidden terminal (because of a collision), the busy tone can not be missed. This exchange also solves the exposed terminal problem, since the receiver does not need to send packets to the transmitter.

This protocol has two clear disadvantages. The first is the need to synchronize time slots between nodes, which is a difficult thing to do. The second is the fact that one busy tone can not be distinguished from another, and therefore a transmitter can mistakenly assume that a certain busy tone was intended for it.

2.4.2. DBTMA. Haas and Deng expanded the concept of busy tones further by using two busy tones in a protocol they named Dual Busy Tone Multiple Access (DBTMA) [10]. The two busy tones allowed the use of an unslotted system and offered more protection against collisions.

The DBTMA mechanism uses an RTS-DATA exchange together with two busy tones. A transaction takes place as follows. The sender transmits an RTS packet and turns on the transmitter busy tone (BT_t). When the RTS is sent, BT_t is turned off. When the receiver receives the RTS, it turns on the receiver busy tone (BT_r). The sender detects the BT_r and sends the DATA packet. The receiver turns off BT_r once the DATA packet is received, and the exchange is complete. A neighbouring node will not send an RTS if either a BT_t or BT_r is detected. Note that since the receiver never sends any packets to the sender, the exposed terminal problem as well as the hidden terminal problem are solved by this protocol.

The solutions provided for the hidden and exposed terminal problems increase throughput dramatically, but like CSMA-CA, fairness properties depend on the actual implementation. The original proposal ([10]) mentions in a footnote that a BEB or MILD backoff algorithm may be used. In the same paper, the authors simulate the MAC protocols DBTMA, RI-BTMA, FAMA, and MACA, and show that their protocol is superior in terms of throughput and delay for several topologies. Note, however, that DBTMA does not include any mechanisms to deal with fairness specifically. A simulation of DBTMA over the topology of figure 2.4, yields fairness results similar to those of DFWMAC. These results are shown in figure 2.6. Again, flow 2 receives over 99% of the throughput.

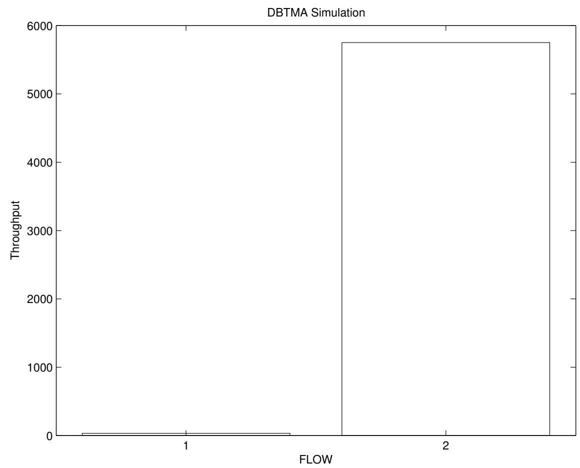

FIG. 2.6. *DBTMA Simulation of the topology in figure 2.4.*

3. Fairness Mechanisms. With the abundance of MAC protocols that are now available, the need for better fairness solutions is more obvious than ever before. Several papers have proposed fairness mechanisms that are not independent protocols, but rather can be incorporated into existing protocols. These mechanisms typically improve fairness properties by taking into consideration information that is already available at each terminal, and using this information to modify the backoff algorithm. These modifications usually add little complexity and little or no messaging overhead, but also result in limited improvements. Problems that require additional information sharing between terminals cannot be solved by these methods since this would require entirely new protocols. Despite this, fairness mechanisms give useful insight into how the fairness problem can be solved. In this section, some of the latest publications on fairness mechanisms are discussed.

3.1. AOB. Bononi et al. proposed a modified backoff algorithm in [6], which they named Asymptotically Optimal Backoff (AOB). This modification is based on p-persistent CSMA and focuses on finding the optimal p. The calculation at each terminal requires only information on how busy the channel is in the vicinity of that terminal. The terminal must first calculate the $Slot\ Utilization$ parameter defined as

$$Slot\ Utilization = \frac{Number\ of\ Busy\ Slots}{Total\ Number\ of\ Available\ Slots}\ .$$

The slot time should be defined by the protocol, for example 802.11b uses a slot time of 20μs. The $Slot\ Utilization$ parameter is simply a measurement of traffic congestion at the terminal. Now p can be calculated as

$$p = 1 - Slot\ Utilization\ .$$

This equation decreases p as the channel becomes busier, which we would expect intuitively. The authors extend this calculation to include a parameter for the number of attempts denoted by NA. NA is equal to 1 on the first transmission attempt, 2 on the second transmission attempt, and so on. This idea gives retransmitted packets a higher probability of being transmitted than new packets. With the addition of this parameter, p is found as follows

$$p = 1 - Slot\ Utilization^{NA}\ .$$

Calculating p in this way gives priority to backlogged terminals.

The authors state that in an ideal network, the channel is not busy 100% of the time and the above equation for p should reflect this fact. They make use of analysis borrowed from [7], which uses an 802.11b network, to show that the ideal slot utilization of the network is independent of the number of nodes, and is strongly dependent on the size of the packets (the authors assume a geometric distribution with parameter q). Since the congestion is dependent only on q, a function can be defined, called Asymptotical Contention Limit or ACL. ACL is the ideal slot utilization in terms of q. The transmission probability p, can now be calculated as,

$$p = 1 - min\left(1, \frac{Slot\ Utilization}{ACL(q)}\right)^{NA}\ .$$

Simulations carried out by the authors show that indeed channel utilization is close to channel capacity even for as many as 200 nodes, while for 802.11b with the BEB backoff algorithm channel utilization drops rapidly as the number of nodes increases. Intuitively, this method is preferable to BEB because the backoff window size is not severely affected by a single collision. Rather, the backoff window size is calculated using congestion information that is measured over a long period of time. However, if the congestion is largely due to hidden terminals, AOB may not perform so well, because it considers $Slot\_Utilization$ at the transmitter and not at the receiver. $Slot\_Utilization$ at the receiver is unknown.

3.2. SBA. The Sensing Backoff Algorithm (SBA) was proposed by Haas and Deng in 2003 [11]. SBA is a backoff algorithm that is intended for wireless LANS in order to make up for the deficiencies of more popular backoff algorithms such as BEB and MILD. SBA works by changing the backoff window size (B) when any of three events occur. When the sender's transmission fails, the backoff window size is multiplied by constant, α, which is larger than 1. When a transmission is successful, the sender and receiver multiply their backoff window size by a constant, θ, which is smaller than 1. Finally, when a node senses a successful transmission in its neighbourhood, it decreases its backoff window size by $\beta \cdot \gamma$, where β is a positive integer and γ is the duration of a single transmission. To summarize, the backoff window is modified according to the following formula:

$$B = \begin{cases} min(\alpha \cdot B, B_{max}) & \text{Transmission failed} \\ max(B - \beta \cdot \gamma, B_{min}) & \text{Sensing another node's successful transmission} \\ max(\theta \cdot B, B_{min}) & \text{Successful transmission at sender and receiver .} \end{cases}$$

B_{max} and B_{min} are the maximum and minimum allowed backoff window sizes respectively. To find the value of the parameters α, β, and θ, the authors analyze a wireless network with N nodes. They make several simplifying assumptions including a fully connected network, and transmissions consisting of only DATA packets and 0 length ACK packets. The result of this analysis yields the optimal backoff window size, $B_{opt} = 4N\gamma$. The values of α, β, and θ must now be chosen so that every node will have a backoff window size of B_{opt} without having to know the value of N. The analysis proceeds as follows. The net change of backoff window sizes of all nodes, $\Delta B_N(t)$, must remain constant in steady state operation. The change in $B_N(t)$ can be divided into two components,

$$\Delta B_N(t) = \Delta B_N^S(t) + \Delta B_N^C(t) \ .$$

$\Delta B_N^S(t)$ is the net change due to successful transmissions and $\Delta B_N^C(t)$ is the net change due to collided transmissions. To simplify the analysis, the authors assume that all nodes have a backoff window size of $\overline{B}$, the average backoff window size of all the nodes. After a successful transmission, the backoff windows of the transmitter and receiver decreases by $2(\theta - 1)\overline{B}$. The backoff windows of the other $N-2$ nodes that will hear the successful transmission will decrease by $-\beta\gamma(N-2)$. Now

$$\Delta B_N^S(t) = \rho^S(t) \cdot [2(\theta - 1)\overline{B} - \beta\gamma(N-2)],$$

where $\rho^S(t)$ is the number of successful transmissions up to time t.

After a collision, the sender increases its backoff window size by $(\alpha - 1)\overline{B}$, and

$$\Delta B_N^C(t) = \rho^C(t) \cdot (\alpha - 1)\overline{B} \ ,$$

where $\rho^C(t)$ is the number of collided transmissions up to time t.

By using the fact that

$$\lim_{t\to\infty} \frac{B_N(t)}{t} = 0,$$

an equation was found relating α, β, θ, and N. The authors chose the values of $\alpha = 1.2$ and $N = 10$ *or* ∞. They claim that these values yield the best results and show some simulation results to prove this. This results in two equations with the two remaining unknowns, β and θ. Solving these equations yields the solution:

$$\alpha = 1.2, \ \beta = 0.8, \ \theta = 0.93 \ .$$

The values of α, β, and θ are also justified in a plot where the authors show that SBA with these parameters performs as well as a network where all nodes have a constant backoff window size of $4N\gamma$. Other plots that compare the throughput of SBA, BEB, and MILD, show that SBA well outperforms MILD. BEB gets much higher throughput than SBA, but the authors claim this is due to the unfairness of BEB. Fairness in this paper is defined as the percentage probability that a node that has just sent a packet will send another packet before any other node. A fairness comparison shows that MILD slightly outperforms SBA, but BEB performs much worse than both.

3.3. Time Based Media Access. Ozugur et al. published several papers ([20][19]) where they proposed a modification to the backoff algorithm based on the number of connections or contention periods between neighbouring nodes. This scheme is also based on p-persistent CSMA and centres around the calculation of the p parameter. In the Connection Based (CB) approach, each station must keep track of the number of connections (i.e. neighbours) that each of its immediate neighbours has. Denoting V_i as the set of all neighbours of i, S_i as the number of neighbours of station i, S_i^{Max}

as the maximum of S_j for $j \in V_i$, then p_{ij} (the p parameter for the flow from station i to station j) is

$$p_{ij} = \begin{cases} 1 & S_i = \sum_{j\in V_i} S_j \\ \min\{1, \frac{S_i}{S_i^{Max}}\} & S_i < \sum_{j\in V_i} S_j \text{ and } S_j = S_i^{Max} \\ \frac{S_j}{S_i^{Max}} & S_i < \sum_{j\in V_i} S_j \text{ and } S_j \neq S_i^{Max} \end{cases} .$$

The first line of this equation states that if all the neighbours of i have no connections other than to i, then $p_{ij} = 1$. The second and third lines state that p_{ij} is a ratio of the "connectivity" of i to j. The second line also adds that if j has more connections than i, then p_{ij} has a maximum of 1. Note that this approach requires stations to periodically broadcast the number of connections they have.

The second approach called the Time Based (TB) approach, requires the periodical broadcast of contention period information. A contention period is the time it takes for a node to acquire the channel (the time between the end of its last transmission and the beginning of its next transmission) denoted by T_{ij}. The paper defines the link descriptor L_{ij} as follows

$$L_{ij} = \begin{cases} 1 & i \text{ had a packet to send to } j \text{ in the last interval} \\ 0 & i \text{ had no packet to send to } j \text{ in the last interval} \end{cases} .$$

p_{ij} is found as follows

$$p_{ij} = min\left\{1, \frac{T_{ij}}{T_{ave}}\right\} ,$$

where,

$$T_{ave} = \frac{\sum_{k\in V_i(k\neq i)}(T_{ki}L_{ki} + T_{ik}L_{ik})}{\sum_{k\in V_i(k\neq i)}(L_{ki} + L_{ik})} .$$

T_{ave} is the average contention period per time interval in the neighbourhood of terminal i.

The authors present simulation results for four different scenarios with four to six nodes. They define two fairness indices. The first is the ratio of maximum station throughput to minimum station throughput and the second is maximum link throughput to minimum link throughput. Simulations over these topologies show very good results, especially for the TB approach. 802.11b with this fairness mechanism achieves 2 to 50 times better fairness than 802.11b without a fairness mechanism.

3.4. Estimation Based Fair Media Access. Bensaou et al. published several articles ([4][3][9]) on adjusting the backoff window size by monitoring the amount of traffic sent by neighbouring stations. In [4] an algorithm is proposed where a terminal calculates the aggregate amount of traffic it sent, denoted by W_i, and the amount of traffic sent by other stations, W_o. If the relative share that flow i should receive is denoted by ϕ_i (e.g. a flow with ϕ=2 should receive double the share of a flow with ϕ=1), then the fairness index, FI, is found by

$$FI = max\left\{\forall\ i,j : max\left(\frac{W_i}{\phi_i}, \frac{W_j}{\phi_j}\right) / min\left(\frac{W_i}{\phi_i}, \frac{W_j}{\phi_j}\right)\right\} .$$

FI must be continuously calculated. The new value for FI is used to update the backoff window size with the following algorithm,

```
switch(FI)
    {
    case >C:
        CW_new = min(CW_old X 2, CWMAX)
    case (1/C,C):
        CW_new = CW_old
    case <1/C:
        CW_new = max(CW_old / 2, CWMIN)
    }
```

where C is a parameter that is close to 1. A C closer to 1 results in more aggressive backoff window adjustments. This algorithm is similar to BEB with the following differences. Adjustments to the backoff window size are made based on long term fairness rather than the results of individual transmissions. This ensures slower adjustments of the window size instead of the large spikes of 802.11. Also, the window size is doubled or halved, rather than doubled or reset.

The authors present simulation results for three scenarios with 3, 4 and 5 nodes, where 802.11 was simulated with and without the fairness mechanism. The results show that fairness is significantly improved.

4. Scheduling. The fairness methods presented in the previous section attempted to dynamically adjust to network congestion by modifying the backoff scheme. The methods described in this section attempt to achieve fairness by using a technique known as *scheduling*. Scheduling is a term that is borrowed from wireline systems. It refers to a system where a schedule is created for processing or transmitting data. If a fair schedule could be created and used by all the nodes of a network, the fairness problem would be solved. Naturally, getting all nodes in a distributed network to follow the same schedule is not a simple task. In this section, three different approaches to scheduling will be examined.

4.1. Fair Queueing. Luo et al. proposed a scheduling approach based on fair queuing (see [17][18]). Fair queuing is used in centralized networks to schedule the processing of multiple flows in a fair manner. Each incoming packet is assigned a start tag and finish tag. The k^{th} packet of flow f is given a start tag S_k^f according to

$$S_k^f = max\{V(A_k^f), F_{k-1}^f\} \ .$$

The start tag of the packet is the arrival time of the packet, $V(A_k^f)$, if there are no packets in the queue, or the finish tag of the last packet in the queue, F_{k-1}^f, if the node is backlogged. The finish tag assigned to the packet, F_k^f, is

$$F_k^f = S_k^f + L_p/r_f \ ,$$

where L_p is the packet length in time and r_f is the relative weight of stream f. If a centralized coordinator was available, packets would be sent in order of lowest start tag to highest. To accomplish this in the distributed ad hoc environment, the start and finish tags of arriving packets must be made known to neighbouring nodes. The authors used an RTS-CTS-DS-DATA-ACK exchange similar to the one used in MACAW. The RTS and CTS packets carry the start tag of the current packet and the DS and ACK packets carry the start tag of the next packet in the queue. Each node must keep a schedule made up of entries of the form $[f, T_f]$. One entry is kept for each flow f. This entry includes the flow id and the latest start tag (T_f). The authors now made three distinct innovations:

Maximizing Local Minimum. Fair queuing requires that the packet with lowest start tag in the network be sent first. This packet cannot be identified because global information is not available at any node. However, the packet with the lowest start tag in the network (absolute minima) is a subset of all the packets that have the lowest start tag in their neighbourhood (local minima). Therefore a node should always transmit a packet if it has the lowest start tag in its schedule.

Backoff mechanism. While all nodes that have a local minimum start tag transmit, additional transmissions may be able to take place without interference. This is known as spatial reuse. To take advantage of spatial reuse, backoff counters can be used by all nodes who do not have the minimum start tag in their neighbourhoods. If the backoff period expires and the channel is idle, a node may assume that none of its neighbours has a local minimum start tag, and may go ahead and transmit. The backoff period must reflect the fact that some nodes have earlier start tags than others. The following backoff window size (B_f) was proposed for flow f with a set of neighbours S:

$$B_f = \sum_{g \in S} I(T_g < T_f) \ mini \ slots \ ,$$

where $I(\cdot)$ is the indicator function. The backoff window is thus proportional to the number of flows that have a smaller start tag than f.

The authors also note that even if the sender has the lowest start tag in its neighbourhood, it might not be the lowest start tag in the neighbourhood of the receiver. Therefore, the receiver must ignore incoming RTS packets that have a higher start tag than other entries in its schedule.

Sliding Window. Spatial reuse can cause a flow to receive more than its fair share of throughput (r_f). To avoid this, a flow can implement a window of size ρ. When the start tags of flow f exceed the window size, f will wait idly for others to transmit.

The authors present simulations of their fair queueing protocol and 802.11 over four topologies with 6 to 35 nodes. In all simulations fairness appears to be significantly improved in the fair queueing simulations. The throughput is usually slightly lower.

4.2. DWOP. Kanodia et al. published a series of papers ([13][12]) on an original MAC protocol named Distributed Wireless Ordering Protocol (DWOP). This protocol tries to achieve fifo fairness by transmitting packets according to their order of arrival. An exchange consists of RTS-CTS-DATA-ACK packets. The RTS and CTS packets contain information on the next packet's time of arrival, while the DATA and ACK packets contain information on the current packet's time of arrival. Every node must maintain a schedule where it logs time of arrival information from overheard transmissions. When a node overhears an RTS or CTS it adds an entry to its schedule. When a node overhears a DATA or ACK packet it deletes an entry from its schedule. A node sends its own packet only if it is first on its schedule. The authors point out two problems with this scheme and propose a solution to each.

Asymmetric Information and Receiver Participation. Because nodes can only keep a schedule of their immediate neighbours, a sender can not know if it has the highest priority in the neighbourhood of its receiver. If the receiver receives a CTS and it is aware that there is a higher priority packet in its neighbourhood, it does not ignore the packet. Rather it sends a CTS and allows the transmission to continue. But to ensure that fairness is restored, the receiver "participates" by adding a notice to the ACK packet. The notice tells the transmitter to backoff by an amount

$$T_{backoff} = R(EIFS + DIFS + T_{success} + CW_{min}) \ ,$$

where $EIFS$, $DIFS$, CW_{min} are short time periods defined by the 802.11 protocol. $T_{success}$ is the time required to carry out an exchange, and R is the rank of the receiver in its own schedule. The term in the brackets is just the overall time required for a complete exchange. This formula ensures that the transmitter backs off long enough for all transmissions with higher priority will take place before the transmitter sends again. This method works effectively in the topology of figure 2.4. Simulation results of the topology shown in figure 4.1 show that DWOP achieves perfect fairness.

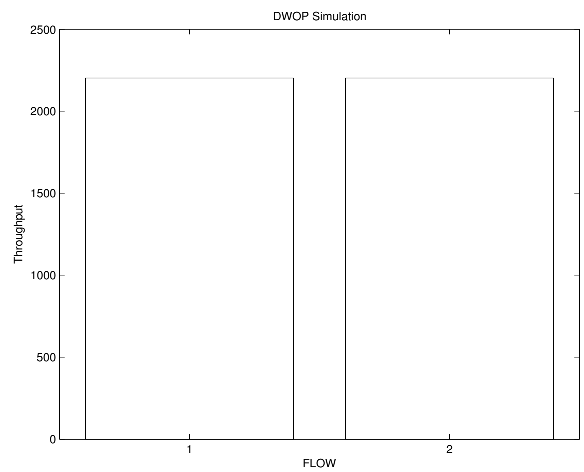

FIG. 4.1. *DWOP Simulation of the topology in figure 2.4.*

Perceived Collisions and Stale Entry Detection. A different problem may occur if a node misses a DATA or ACK packet. This can happen when two transmissions take place around the same node at the same time, and is referred to as a perceived collision (since it's not really a collision but the node is unable to extract schedule information from either transmission nonetheless). The node will then fail to remove an entry from its schedule. The node will wait indefinitely for this packet to be transmitted and will not send its own packets. To solve this, the authors introduce stale entry removal, whereby a node detecting a new entry for a given flow, may assume that all previous entries are stale and can be deleted. This way, fairness may be compromised, but only for a very short period of time, until the flow whose transmission was missed sends again.

In order to keep the schedules updated, DWOP requires all nodes to be continuously backlogged. The authors present simulations of DWOP and 802.11 over three topologies containing 4, 6, and 10 nodes. In all simulations, fairness was drastically improved, but throughput decreased, sometimes significantly. This is due to the fact that DWOP maintains fifo schedules strictly without spatial reuse. A simulation of

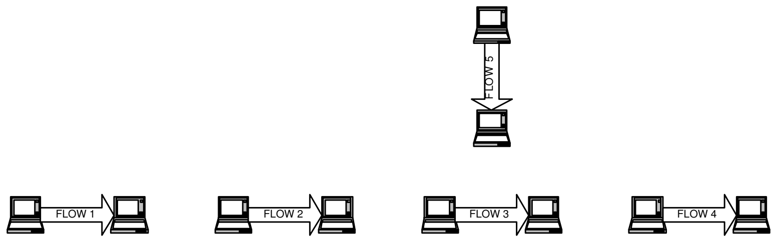

FIG. 4.2. *Network topology with 10 nodes and 5 flows.*

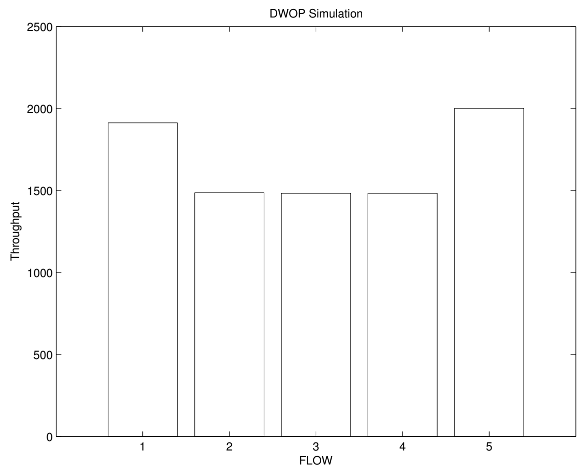

FIG. 4.3. *DWOP Simulation results for the topology in figure 4.3*

DWOP with the topology in figure 4.2 demonstrates this. The results are shown in figure 4.3. The capacity of the channel is approximately 5000 packets, so ideally each flow should send about 2500 packets. However, flow 1 and 5 send less than 80% of this figure and flows 2, 3, and 4, send less than 60% of this figure. The authors also evaluated the short term fairness of their protocol. Short term fairness was measured by the number of consecutive packets sent by a terminal before any other terminal captured the channel. The authors demonstrated good improvements in short term fairness of DWOP over 802.11 in all topologies.

4.3. RRMS. Randomly Ranked Mini Slots (RRMS) makes use of Pseudo Noise (PN) sequences [8]. Each node generates a unique sequence with a random number generator such as a shift register (see [21] for more details). Each node uses a random seed to initialize its random number generator, ensuring that each PN sequence is unique. Nodes must occasionally broadcast their seed and the seed of all their one hop neighbours, so that every node can keep track of all the PN sequences in its two hop neighbourhood. Time is divided into slots that are synchronized between neighbours. Each number in the PN sequence is associated with a time slot. The number in the PN sequence associated with the current time slot is called the *rank*. Transmitters only send an RTS in the current time slot if their rank is higher than the ranks of all the transmitters of interfering flows. Since the probability that the rank of a given transmitter is higher than all interfering transmitters is a function of the number of interfering transmitters alone (since the PN sequences are uncorrelated),

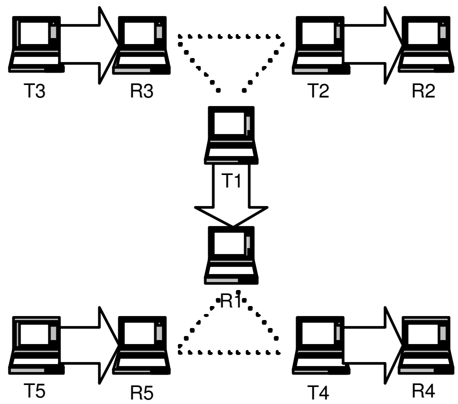

FIG. 4.4. *T1 must consider T4's and T3's ranks when sending an RTS. The other nodes do not interfere with T1's transmission.*

the fraction of channel capacity that any transmitter is allocated depends only on the number of interfering flows and not on the specific topology.

To take advantage of spatial reuse time slots are divided into mini slots. Whereas traditionally a slot is the length of time required to transmit a complete exchange, RRMS' exchange takes place over multiple slots. A mini slot length is set long enough to contain the RTS packet plus overhead.

To achieve the highest possible throughput, an RTS-DATA exchange is used together with a receiver busy tone. A transaction begins with the sender sending an RTS packet in the first mini slot. The receiver on hearing the RTS, transmits a busy tone. The sender detects the busy tone and sends the DATA packet starting at the second mini slot. Once the DATA packet is received (a number of mini slots later), the receiver terminates the busy tone. Since the receiver never sends packets, this exchange solves the exposed terminal problem. A transmitter may send an RTS in the current time slot if a busy tone is not detected, and its rank is higher than all transmitters that are one hop away from its receiver and its rank is higher than all transmitters whose receiver is one hop away from itself (see figure 4.1). This ensures two interfering transmitters will not transmit simultaneously.

While the mini slot concept utilizes spatial reuse well and improves throughput, it also degrades fairness considerably. To see this, consider the topology in figure 1.2 and assume an exchange lasts for N mini slots. Suppose that in the first mini slot C has a higher rank than B and B has a higher rank than A. Node A can capture the channel from the 2^{nd} to N^{th} mini slot, while C is still transmitting. In fact A will capture the channel while C is transmitting with a probability of $1-(\frac{1}{2})^{N-1}$ (e.g. 99.8% for $N=10$). When C completes its exchange, B will not be able to transmit since A will be transmitting. Thus C will have a better chance of capturing the channel than B. B's share of throughput in this topology is very small. This problem is solved by introducing rank attenuation. After a node completes a transaction, it sets its own rank to 0 for the next N mini slots. After N mini slots, the rank returns to the value it would have had if rank attenuation was not used. This ensures that every node, even one that is in a congested neighbourhood, will receive a fair share of throughput.

In [8], RRMS is evaluated in terms of throughput, long term fairness and short term fairness in comparison with DBTMA. Short term fairness is defined as the Root

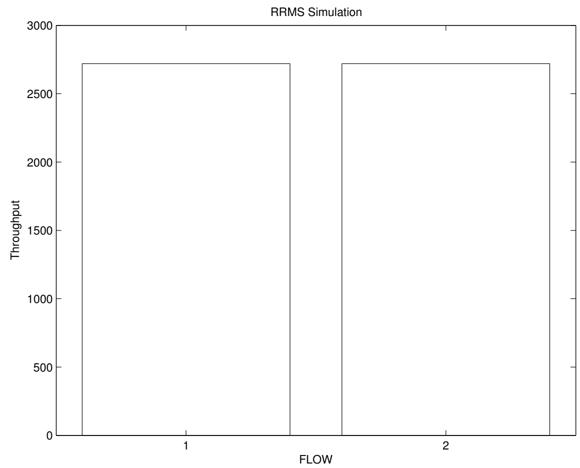

FIG. 4.5. *RRMS Simulation results for the topology in figure 2.4.*

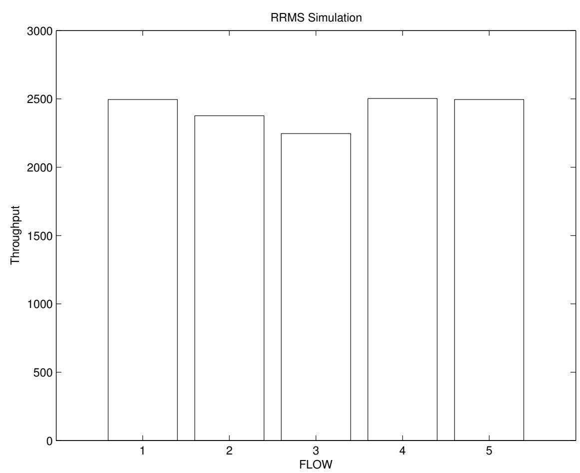

FIG. 4.6. *RRMS Simulation results for the topology in figure 4.6. RRMS takes good advantage of spatial reuse.*

Mean Square (RMS) deviation of the simulated delivery time sequence (the times packets arrived at the receiver) from an ideal delivery time sequence. An ideal sequence is the sequence that would have resulted if a central coordinator was available. Simulations over topologies including random topologies with 100 nodes show that RRMS yields excellent fairness properties and good throughput results. A simulation of RRMS with the topology of figure 2.4 shows that RRMS does achieve perfect fairness and high throughput. The results are shown in figure 4.5. Another simulation with the topology of figure 4.2, demonstrates that RRMS takes good advantage of spatial reuse. These simulation results are shown in figure 4.6. The maximum throughput that can be achieved by a node that captures the channel for the entire simulation time is approximately 5000 packets. Note that every transmitter in figure 4.6 sends a number of packets close to the ideal 2500.

5. Summary. The mechanisms and protocols described in this chapter offer a multitude of approaches to achieve fairness at the MAC layer of ad hoc networks. In this section we will enumerate the fairness techniques and compare their main advantages and disadvantages.

5.1. AOB. The AOB mechanism is simple to implement and can easily fit within the framework of other MAC protocols such as 802.11. Both $Slot\ Utilization$ and NA (number of attempts) can be found easily. The q parameter (distribution of packet length) may or may not be easy to determine. The effectiveness of this mechanism has not been demonstrated for complex topologies.

5.2. SBA. The SBA mechanism requires nodes to adjust their backoff window sizes according to a straight forward algorithm that is easy to implement. While its authors have shown that it works ideally in a fully connected network, its operation in a multi hop network requires further investigation. This is largely due to the fact that nodes may not be aware of successful transmissions around them because of interfering transmissions and will not be able to adjust their backoff windows as needed.

5.3. Time Based Media Access. The two mechanisms, CB and TB, both require periodic broadcasting of information. This information may be quite large in the case of TB, since its length is proportional to the number of neighbours a node has. The effect of nodes not hearing this information (due to interference) from their neighbours is unclear. The authors presented good results based on their definition of fairness. However, this definition may not be a good benchmark for some scenarios.

5.4. Estimation Based Fair Media Access. This mechanism requires estimating the amount of data sent by every other flow. This may be very hard to do in some topologies, due to interfering transmissions. If traffic arrivals are frequent and the channel is heavily utilized, nodes may not overhear their neighbours' transmissions at all. Also, attempting to divide throughput according to the various ϕ_i values may be unreasonable because of the topology.

5.5. Fair Queueing. The fair queueing protocol proposed by Luo is an extensive protocol that uses several ideas to maintain fairness according to predefined rates (r_f). Nodes must overhear neighbours' transmissions to maintain their schedules, but the system can tolerate occasional missed updates. The fact that the sender's schedule does not take into account the receiver's neighbourhood remains a problem. It is unclear whether the backoff mechanism is a good solution for this problem. The authors simulated each of their ideas independently. Some ideas were beneficial in some scenarios and detrimental in others. More careful study of each of the ideas is necessary to form a complete protocol that works in all scenarios.

5.6. DWOP. This protocol also requires periodic broadcasts to keep schedules updated. The authors demonstrated the good fairness properties of their protocol. However, there are no provisions to take advantage of spatial reuse, which makes an obvious impact on the results. The receiver participation concept offers an original and effective idea to deal with the sender's lack of knowledge of the receiver's neighbourhood.

5.7. RRMS. The RRMS protocol includes several innovative ideas that result in very good throughput and fairness properties. A small amount of information must still be broadcast between nodes. However, if the broadcast was heard once from a given node, future broadcasts may be missed with little to no effect. Time slots must also be synchronized between nodes, but perfect synchronization is not necessary. The protocol performs very well in widely differing topologies.

5.8. Performance Comparison. To give the reader an idea of how the different protocols compare, simulations results for DFWMAC, DBTMA, DWOP, and RRMS are presented here. Each simulation lasted for 20 seconds and used a topology of 100 nodes randomly distributed in a square area with edge wrapping. The transmission range was adjusted so that each node had an average of 6 neighbours. The probability of each node being a transmitter was 1/6. Each transmitter randomly chose one of its neighbours to be a receiver in the beginning of the simulation. The traffic arrival model at each transmitter was a Poisson process with parameter $\lambda = \frac{1}{10^i}$ with units of packets/ms and i varying from 1.0 to 2.0 in steps of 0.2. These values of λ were used because when $\lambda = \frac{1}{10}$ packets/ms the network is heavily congested (since the length of an exchange is approximately 10ms), and when $\lambda = \frac{1}{100}$ packets/ms the network has virtually no congestion.

30 simulations were run for each value of lambda with each protocol. The throughput of every group of 30 runs was averaged. The results with 90% confidence intervals are shown in figure 5.1(a). To evaluate fairness we first found the ideal delivery sequence, S_{ideal}. This is the sequence that would result if a central coordinator was available. We formed this sequence with the following algorithm.

```
TIME = TxTIME
while(NOT ALL PACKETS HAVE BEEN ADDED TO THE SEQUENCE)
    ADD packet with smallest time of arrival

    while(SOME PACKETS CAN STILL BE ADDED IN THIS TIME SLOT)
        ADD packet with smallest time of arrival that
            can be sent in this time slot without interference
    end loop

    TIME = TIME + TxTIME
end loop
```

`TxTIME` is the time required for a single transmission. This number is the sum of durations of all the packets in an exchange and the expected value of the backoff algorithm when there are no collisions. The ideal sequence is simply a fifo sequence with spatial reuse.

The simulated sequence, S_{sim}, is found via the simulation as described above. The number of packets sent by flow i in the ideal and simulated sequences is denoted by $S_{ideal}(i)$ and $S_{sim}(i)$ respectively. The Root Mean Square Error (RMSE) between the normalized ideal sequence and the normalized simulated sequence is given by

$$RMSE = \sqrt{\sum_{\forall i} \left[\left(\frac{S_{ideal}(i)}{|S_{ideal}|} \right)^2 - \left(\frac{S_{sim}(i)}{|S_{sim}|} \right)^2 \right]},$$

where $|S_{ideal}|$ and $|S_{sim}|$ are the total throughputs of the ideal and simulated sequences respectively. Note that as a result of the normalization by the total throughput, this fairness measure is not biased by the throughput of the simulation and is a measure of fairness only. RMSE results were averaged over 30 trials and are shown with 90% confidence intervals in figure 5.1(b) for different values of λ.

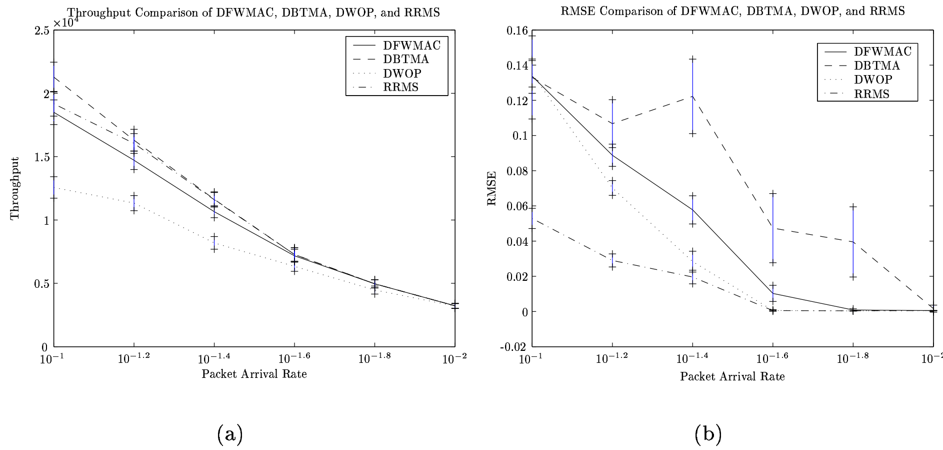

FIG. 5.1. *Simulation results for DFWMAC, DBTMA, DWOP, and RRMS for varying λ, with 90% confidence intervals.*

6. Conclusions. The future success of ad hoc networks relies on finding fair, efficient, and robust MAC layer algorithms and protocols. A good MAC layer protocol would have to maximize throughput and allocate it fairly among contending flows. Several existing MAC protocols have dealt with the hidden and exposed terminal problems – a major cause of throughput and fairness degradation. Other methods were developed to take advantage of available information to improve backoff algorithms. More complicated, distributed algorithms have also been developed to increase information sharing among terminals to create fair transmission schedules. These are known as scheduling algorithms. Future MAC standards should combine all of these ideas to support the large throughput users have come to expect from their applications.

REFERENCES

[1] *IEEE Standard, ANSI/IEEE Std 802.11, 1999 Edition, Part 11*, 1999.
[2] N. ABRAMSON, *The Aloha System – Another Alternative for Computer Communications*, in Proc. Fall Joint Comput. Conf., p37, 1970.
[3] B. BENSAOU AND Y. WANG, *Achieving fairness in IEEE 802.11 DFWMAC with variable packet lengths*, in Proc. IEEE Global Telecommunications Conference (GLOBECOM), 2001, pp. 3588–3593.
[4] B. BENSAOU, Y. WANG, AND C. C. KO, *Fair medium access in 802.11 based wireless ad-hoc networks*, in Proceedings of the first ACM international symposium on Mobile and ad hoc networking & computing (MOBIHOC), IEEE Press, 2000, pp. 99–106.
[5] V. BHARGHAVAN, A. DEMERS, S. SHENKER, AND L. ZHANG, *MACAW: A Media Access Protocol for Wireless LANs*, in Proc. ACM SIGCOMM, 1994, pp. 212–225.
[6] L. BONONI, M. CONTI, AND E. GREGORI, *Design and Performance Evaluation of an Asymptotically Optimal Backoff Algorithm for IEEE 802.11 Wireless LANs*, in Proceedings of the 33rd Hawaii International Conference on System Sciences, 2000.
[7] F. CAL, M. CONTI, AND E. GREGORI, *IEEE 802.11 Wireless LAN: Capacity Analysis and Protocol Enhancement*, in Proceedings of the 17th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), 1998, pp. 142–149.
[8] J. ESHET AND B. LIANG, *The RRMS Protocol: Fair Medium Access in Ad Hoc Networks*, in 22nd Biennial Symposium on Communications, Queen's University, 2004.
[9] Z. FANG, B. BENSAOU, AND Y. WANG, *Performance evaluation of a fair backoff algorithm for*

IEEE 802.11 DFWMAC, in Proceedings of the third ACM international symposium on Mobile ad hoc networking & computing (MOBIHOC), ACM Press, 2002, pp. 48–57.

[10] Z. Haas and J. Deng, *Dual busy tone multiple access (DBTMA)-a multiple access control scheme for ad hoc networks*, IEEE Trans. on Commun., 50 (2002).

[11] ———, *On optimizing the backoff interval for random access schemes*, IEEE Trans. on Commun., 51 (2003), pp. 2081–2090.

[12] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly, *Distributed multi-hop scheduling and medium access with delay and throughput constraints*, in Proceedings of the 7th annual international conference on Mobile computing and networking (MOBICOM), ACM Press, 2001, pp. 200–209.

[13] V. Kanodia, A. Sabharwal, B. Sadeghi, and E. Knightly, *Ordered packet scheduling in wireless ad hoc networks: mechanisms and performance analysis*, in Proceedings of the third ACM international symposium on Mobile ad hoc networking & computing (MOBIHOC), ACM Press, 2002, pp. 58–70.

[14] P. Karn, *MACA – A New Channel Access Method for Packet Radio*, ARRL/CRRL Amateur Radio 9th Computer Networking Conference, September 1990.

[15] A. Leon-Garcia and I. Widjaja, *Communication Networks*, McGraw-Hill, 2000.

[16] J. Li, C. Blake, D. S. D. Couto, H. I. Lee, and R. Morris, *Capacity of ad hoc wireless networks*, in Proceedings of the seventh annual international conference on Mobile computing and networking (MOBICOM), ACM Press, 2001, pp. 61–69.

[17] H. Luo, S. Lu, and V. Bharghavan, *A new model for packet scheduling in multihop wireless networks*, in Proceedings of the sixth annual international conference on Mobile computing and networking (MOBICOM), ACM Press, 2000, pp. 76–86.

[18] H. Luo, P. Medvedev, J. Cheng, and S. Lu, *A self-coordinating approach to distributed fair queueing in ad hoc wireless networks*, in Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), 2001, pp. 1370–1379.

[19] T. Ozugur, *Optimal MAC-Layer Fairness in 802.11 Networks*, in Proc. IEEE International Communications Conference (ICC), 2002.

[20] T. Ozugur, M. Naghshineh, P. Kermani, and J. A. Copeland, *Fair Media Access for Wireless LANs*, in Proc. IEEE Global Telecommunications Conference (GLOBECOM), 1999.

[21] R. Rozovsky and P. R. Kumar, *SEEDEX: a MAC protocol for ad hoc networks*, in Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing (MOBIHOC), ACM Press, 2001, pp. 67–75.

[22] F. A. Tobagi and L. Kleinrock, *Packet switching in radio channels: part II the hidden terminal problem in carrier sense multiple-access and the busy-tone solution*, IEEE Trans. on Commun., COM-23 (1975), pp. 1417–1433.

[23] C. Wu and V. Li, *Receiver-initiated busy-tone multiple access in packet radio networks*, in Proceedings of the ACM workshop on Frontiers in computer communications technology, ACM Press, 1988, pp. 336–342.