Isolation of Wireless Ad hoc Medium Access Mechanisms Under UDP

Mario Correa, Ken Tang, Mario Gerla
Computer Science Department
University of California, Los Angeles
{mcorrea, ktang, gerla}@cs.ucla.edu

Abstract

Mobile computing is the way of the future, as evident by such initiatives as Bluetooth, Iceberg and HomeRF. However, for mobile computing to be successful, an obvious layer, the MAC layer, must be efficient in channel access and reservation. Therefore, in-depth understanding is needed of the wireless MAC layer if wireless computing is to takeoff. Many random access wireless MAC protocols have been proposed and even standardized. However, there has yet been an attempt to understand why certain designs are used and what makes certain protocols better than others. In this paper, we survey several popular, contemporary, wireless, random access MAC protocols and determine the effects behind the design choices of these protocols.

1. Introduction

Mobile computing has proliferated in recent years. Initiatives, such as Bluetooth [Haartsen98], Iceberg [Joseph98] and HomeRF [Negus98] are making wireless networking a reality. Such technologies allow for wireless communications between numerous devices, such as the PC, PDA, cellular and cordless phones, pagers and other appliances, to be seamless and practical. With such diverse and dense communication environment, controlling access to the wireless channel plays an integral part in making Bluetooth, Iceberg and HomeRF become a reality. Therefore, the success of mobile computing technologies will in part hinge on the effectiveness of the MAC layer.

In this paper, we provide insights into the design of contemporary wireless, ad hoc random access MAC layer protocols. In particular, we focus on the various methods of channel access used to resolve contention on the wireless medium. Then we isolate the individual medium access component through experiments to determine the effect each component has on the network.

In section 2, a background of wireless MAC layers is introduced. Section 3 discusses our research study. Simulation experiments are described in section 4. Results from our UDP experiments under various MAC protocols are presented in section 5. Section 6 concludes the paper. Finally, the Appendix reveal the results of the simulation experiments conducted regarding constant bit rate with UDP in the form of tables.

2. Background

Although many MAC layer protocols have been developed, understood and analyzed since the early 1970s, these MAC layers were associated mainly with a wired network. In this section, we present the difference characteristics between a wireless versus a wired environment and argue that traditional wired MAC layer protocols do not work well in a wireless environment.

2.1. Wireless Medium versus Wired Medium

The characteristics of a wireless medium differ from that of the traditional wired networks. These dissimilarities include the limited range and battery power, high bit-error rates, mobility, difficulty in collision detection, "hidden terminal" and "exposed terminal" problems. The limited range and battery power of mobile nodes result in individual mobile nodes having different send/received signal power. High bit error rates are caused by external noise that can distort or interfere with the transmitted signal. Unlike wired nodes, wireless nodes are expected to be on the move frequently, therefore making certain schemes such as Token Ring extremely difficult, if not impossible. Collision detection, although simple in a wired environment, is extremely difficult to achieve in a wireless environment. With one antenna, a mobile node can only listen or send data, but not both simultaneously. Full-duplex radios that allow the mobile node to send and receive data at the same time are extremely expensive and thus are not a practical solution. Other differences between a wired and wireless network are that the "hidden terminal" and "exposed terminal" problems exist in a wireless network. In a wired network, these two problems are non-existent. The "hidden terminal" problem illustrates the difference that collision of data packets occurs at the receiver, rather than at the sender. Even if the channel is free within the sender's range, it may not be free at the receiver. In the "exposed terminal" problem, although the medium is sensed busy near the transmitter, the medium maybe free near the intended receiver. Figure 1 illustrates the "hidden terminal" and "exposed terminal problems. These differences between wireless and wired networks make medium access in wireless networks inherent more difficult and complex compared to that of its wired counterpart.

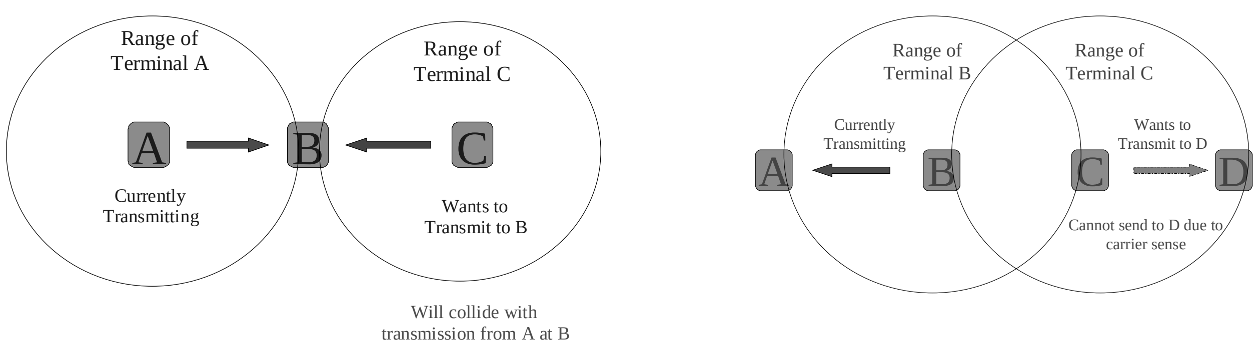

Figure 1. (Left) Relying on carrier sense, Station C decides to transmit to B, but that transmission collides with the transmission currently underway from Station A to Station B. (Right) Relying on carrier sense, Station C postpones a transmission to Station D, even though this transmission would not collide with the one underway from Station B to Station A.

2.2. Existing Wireless Random Access MAC Schemes

This section presents an overview of several popular random access wireless MAC layer protocols. The protocols include Carrier Sense Multiple Access (CSMA), Multiple Access with Collision Avoidance (MACA), MACAW, Floor Acquisition Multiple Access (FAMA) and IEEE 802.11.

CSMA (Carrier Sense Multiple Access) requires carrier sensing before transmission. If the channel is free, the packet is transmitted immediately. Otherwise, it is rescheduled after a random timeout. The major limitation of CSMA is the "hidden terminal" and "exposed terminal" problem [Tanenbaum96]. CSMA was used first in the Packet Radio network in the mid 1970's [Jubin87].

An alternative to this scheme involves channel reservation overhead in order to reduce the number of collisions. This scheme, called Multiple Access with Collision Avoidance (MACA), utilizes control frames, which can be overheard by stations with the ability to collide with the upcoming transmission [Karn90]. These control frames are of fixed size, and also contain the length of the proposed data transmission. The station wanting to transmit initiates a Request To Send (RTS) control frame, and sets a timer to await the Clear To Send (CTS) control frame from the intended receiver. Any station overhearing the RTS control frame is within range to collide with the expected CTS frame, and must defer transmission for a period of time long enough to ensure delivery of the CTS. If the station does not receive the CTS within a specified timeout interval, it assumes a collision has occurred, and initiates a backoff before attempting to retransmit. Any station overhearing the CTS control frame is within range to collide with the upcoming transmission, and is required to defer for the full duration of transmission, as specified in the control frame. In Figure 1 Left, Station C would overhear this CTS frame from Station B, and defer transmission. In Figure 1 Right, Station C will not overhear this CTS frame from Station A, and will be free to initiate transmission with Station D after the expected roundtrip time for the CTS frame.

MACAW [Bharghavan94] is an enhancement to MACA. MACAW utilizes a MILD backoff to select the retransmission time. Besides using RTS/CTS, three other control frames are introduced (DS, ACK, RRTS). The protocol also implements a backoff copying scheme that allows stations to use the same backoff counters to when transmitting to the same station. Moreover, a multiple stream model is included to foster fairness.

Floor Acquisition Multiple Access (FAMA) is an experimental MAC protocol specifically developed for the Glomo DARPA program [Fullmer95]. It is a descendent of MACA. Like MACA, it uses RTS and CTS exchange to prepare the floor (medium) for data transmission. However, unlike MACA, it also utilizes carrier sensing.

IEEE 802.11 [Editors96] expands on FAMA by adding link-level ACKs in addition to the RTS/CTS control frames. Furthermore, 802.11 abandons the traditional CSMA in favor of CSMA/CA (Collision Avoidance). In 802.11, the Distributed Coordination Function (DCF) represents the basic access method that mobile nodes utilize to share the wireless channel. The scheme incorporates CSMA with Collision Avoidance (CSMA/CA) and acknowledgement (ACK). Optionally, the mobile nodes can make use of the virtual carrier sense mechanism that employs RTS/CTS exchange for channel reservation and fragmentation of packets in situations where the wireless channel experiences high bit error rate. CSMA/CA works as follows: a node wishing to transmit senses the channel, and if it is free for a time equal to the DCF InterFrame Space (DIFS), the node transmits. If the channel is busy, the node enters a state of collision avoidance and backs off from transmitting for a specified interval. In the collision avoidance state, the node sensing the channel busy will suspend it's backoff timer, only resuming the backoff countdown when the channel is again sensed free for a DIFS period. A typical sequence of exchange in 802.11 using the virtual carrier sensing mechanism involves the source node first sensing the channel using CSMA/CA. After CSMA/CA is executed, the source node transmits a RTS, followed by the destination node responding with a CTS, then with the source node sending the data frame and ending with the destination node confirming with an ACK to the source node.

3. Research Goals

The various random access schemes for wireless ad hoc networks presented in section 2.2 employ different methods, or components, for channel contention on the wireless medium. Among them are carrier sense, packet sense (no carrier sense), carrier sense with collision avoidance, RTS/CTS control frames and link-level ACKs. For example, CSMA solely uses carrier sense. FAMA utilizes carrier sense with RTS/CTS control frames. MACA, on the other hand, digresses from the carrier sensing of FAMA and instead opts for packet sensing. MACAW adds on top of MACA link-level ACKs, among other features. Finally, 802.11 (with virtual carrier sense) coalesces FAMA, link-level ACKs and collision avoidance together.

In our study, we isolate each of the above features to determine the individual effect each component incurs on network performance. The metrics of interest is throughput and fairness. To separate the components, we implemented various combinations of the above features and compare them among one another. The protocols implemented are packet sense (PSMA), carrier sense (CSMA), carrier sense with collision avoidance (CSMA/CA), PSMA/RTS/CTS, CSMA/RTS/CTS, CSMA/CA/RTS/CTS, PSMA/ACK, CSMA/ACK, CSMA/CA/ACK, PSMA/RTS/CTS/ACK, CSMA/RTS/CTS/ACK and CSMA/CA/RTS/CTS/ACK. The simulation platform used is GloMoSim [Zeng98]. GloMoSim is a parallel simulation environment implemented in PARSEC, PARallel

Simulation Environment for Complex Systems [Bagrodia98]. It includes various wireless protocols in its library (radio propagation, mobility, MAC, network, transport and applications). In addition, GloMoSim provides a valuable and useful feature that facilitates different protocols at a given layer to be swapped in and out of the protocol stack and thus allows for comparison between these different protocols. Most importantly, GloMoSim permits the detailed modeling of several layers and the study of their interaction, yet preserving very good runtime efficiency and yielding manageable execution time.

4. Experimental Configuration and Parameters

For our simulation experiments, we consider several topologies (Figure 2 – Figure 7): string, hidden terminal, exposed terminal, ring, and bottleneck. The arrows represent the direction of data packet transmissions. Constant bit rate (CBR) with unbounded backlog executing with UDP is used for the application. We utilize static routing to route packets since mobility is not considered and use Bellman-Ford routing when mobility is introduced. Different MAC protocols are evaluated: PSMA, CSMA, CSMA/CA, PSMA/RTS/CTS, CSMA/RTS/CTS, CSMA/CA/RTS/CTS, PSMA/ACK, CSMA/ACK, CSMA/CA/ACK, PSMA/RTS/CTS/ACK, CSMA/RTS/CTS/ACK and CSMA/CA/RTS/CTS/ACK. Radios with no capture ability are modeled with a channel bandwidth of 2Mbps. Furthermore, the channel uses free-space with no external noise (perfect channel).

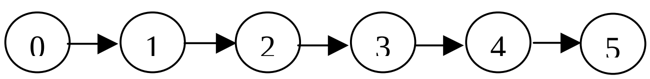

Figure 2. String Topology.

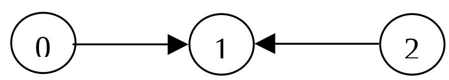

Figure 3. Hidden Terminal Topology.

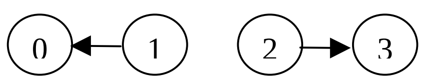

Figure 4. Exposed Terminal Topology

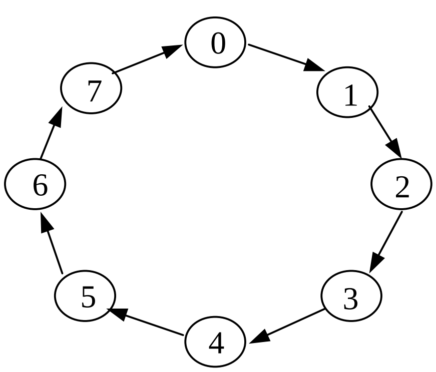

Figure 5. Ring Topology.

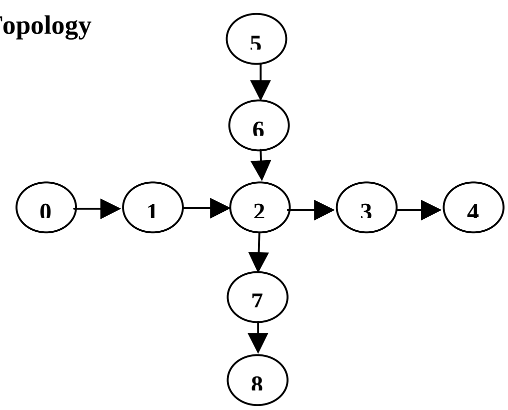

Figure 6. Cross Topology

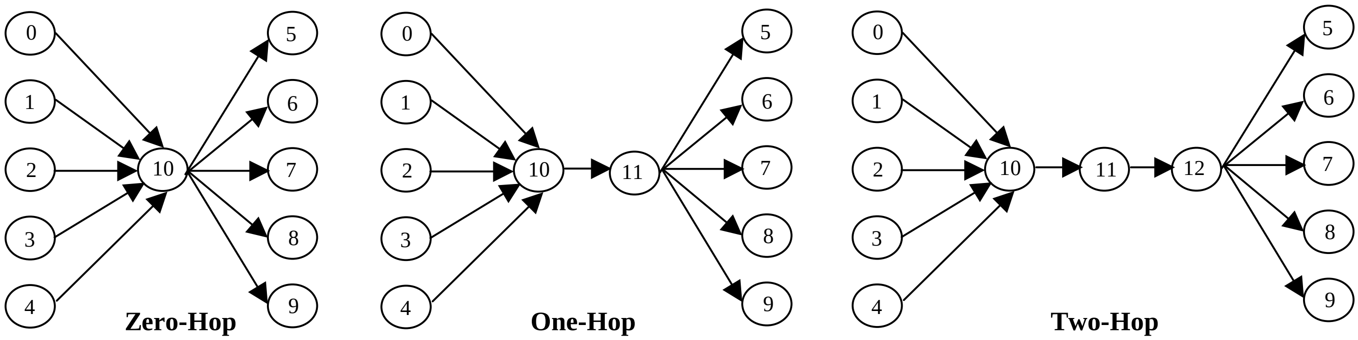

Figure 7. Bottleneck Topology.

Each node has a 25-packet MAC layer buffer pool. Scheduling of packet transmissions is FIFO. Each simulation run is executed for 200 simulated seconds.

Seven different sets of experiments are examined. First, with the variable number of hops experiment, we vary the number of hops for a connection, from 1 to 5 using the topology in Figure 2. Each node is only within range of their intermediate neighbors. Next, we wish to examine the behavior of a system that combines end-to-end and intermediate sessions. We consider the case of a connection spanning 5 hops sharing a path with another flow that occupies a shorter number of hops along the same path. Thus, from Figure 2, we have an end-to-end connection from node 0 to node 5 and a single intermediate flow from either node 2 to node 3 or node 1 to node 4. In other words, the intermediate session varies between one and three hops. Again, each node is only within range of their intermediate neighbors. We next divert our attention to the hidden terminal environment depicted in Figure 3. Node 1 is in radio range of node 0 and node 2. Node 0 and node 2 cannot are not within reception range of each other. Connections are set up from node 0 to node 1 and from node 2 to node 1. With the exposed terminal experiments shown in Figure 4, node 0 is in range of node 1, node 3 is in range of node 2, node 1 is in range of both node 0 and node 2 and node 2 is in range of both node 1 and node 3. Two connections are established, one between node 1 and node 0 and the other between node 2 and node 3. Then, we consider a ring topology as depicted in Figure 5. The 8 nodes are engaged in single hop connections (node 0 to node 1, node 1 to node 2, … , node 7 to node 0) with each node being able to reach its neighboring nodes only. Next, we turn our attention to streams that intersect one another, the cross experiments. The configuration consists of 9 nodes forming a cross, as shown in Figure 6. Two sets of experiments are conducted. The first set consists of a single session from node 1 to node 3 (two-hop) competing with another single session from node 6 to node 7 (two-hop). The second set is composed of a single session from node 0 to node 4 (three-hop) contending with a session from node 5 to node 8 (three-hop). Here also a node is only aware of its neighboring nodes. Finally, we examine multiple flows sharing a common bottleneck, as depicted in Figure 7. Here, we have connections from node 0 to node 5, node 1 to node 6, node 2 to node 7, node 3 to node 8 and node 4 to node 9. All sessions travel through common nodes and links, with the number of hops for the bottleneck link varying from 0 to 2. Furthermore, the senders (node 0, 1, 2, 3 and 4) are within transmission range of each other. The receivers (node 5, 6, 7, 8 and 9) are also within range of each other.

5. Constant Bit Rate with UDP

To investigate the impact of the individual medium access components (PSMA, CSMA, CSMA/CA, RTS/CTS and ACK) under UDP, we evaluate various protocols using the experiments mention in section 4.

5.1. PSMA, CSMA, and CSMA/CA

The throughput for a single CBR client/server pair over a variable number of hops is inversely proportional to the number of hops. For the single hop case PSMA and CSMA have comparable throughputs and both have better throughput than CSMA/CA. Table 1 shows that CSMA and CSMA/CA have a greater throughput than PSMA as the number of hops increases. This is expected since PSMA sends without sensing the medium and results in poor performance when the destination is more than one hop away. Collision avoidance virtually has no impact for this series of simulations because there is no feedback from the frame loss and hence the timer backoff values never increase. Instead, the extra wait time slightly reduces the overall throughput when compared to CSMA.

Next we consider the situation the end-to-end vs. intermediate connections experiments where there are two client/server pairs sending to each other in the string topology. In the first case, when the clients cannot hear each other. Comparison of PSMA, CSMA, and CSMA/CA throughput results, Table 2, shows that the performance of the first two protocols is comparable and that both have a greater throughput than CSMA/CA. Observe that one pair captures the channel regardless of the protocol. In effect, this experiment is reduced to the one-hop single CBR client server pair. When both clients can hear each other, as in the second string experiment, the throughput for the three protocols is less than their throughput in the first string experiment. This is to be expected since PSMA's constant sending results in more collisions. Also, CSMA and CSMA/CA sense the channel busy more often than in the first case so they are not able to send as often. Unlike the first case, CSMA performs better than both PSMA and CSMA/CA. It is a bit more cautious in sending than PSMA and does not pay the penalty of the extra waiting before sending that CSMA/CA does. Again, only one client/server pair succeeds in communicating.

Table 3 shows the poor performance of PSMA, CSMA, and CSMA/CA in the hidden terminal experiments. The 2-1 pair captures the channel, preventing the 1-0 pair from ever communicating. There is not much difference in performance of the three schemes. Observe that collision avoidance does not work well in this scenario.

The most aggressive protocols with the least overhead do the best in the exposed terminal topology. As shown in Table 4, PSMA does exceedingly well with a throughput of 1.9Mbps which close to the 2Mbps channel capacity.

CSMA is only able to utilize 95% of the channel capacity and only the 2-3 pair is able to communicate. The extra waiting time imposed by CSMA/CA severely limits its throughput to approximately half the channel capacity.

The three protocols perform poorly in the ring experiments with respect to aggregate throughput, Table 5. CSMA/CA has the advantage that all the senders are able to establish a connection and send something, however small, to the receivers. The 7-0 pair is the only pair that has non-negligible throughput for all the protocols. In an environment with many senders and receivers, none of the protocols has an effective mechanism for coordinating transmissions between nodes. We expect to see similar performance for the more complex bottleneck topologies.

Looking at the first cross traffic experiment, Table 6, we see that PSMA, CSMA, and CSMA/CA all perform poorly with CSMA faring the worst. Not only is the throughput of three schemes as poor as in the hidden terminal experiments but also one pair captures the channel. For the second experiment, the performance of CSMA and CSMA/CA improve while the performance of PSMA practically remains unchanged.

Tables 7-9 show that PSMA, CSMA, and CSMA/CA all do fairly poorly in the bottleneck topologies. In the zero-hop bottleneck, all of the CSMA and CSMA/CA clients are able to successfully send a few packets to servers. Only one PSMA client is able to successfully transmit packets. The situation degrades as the number of hops in the bottleneck increases. CSMA and CSMA/CA fare a bit better overall with respect to aggregate throughput.

PSMA performs the best in single hop and exposed terminal situations. CSMA and CSMA/CA perform comparably in most situations. CSMA/CA performs slightly better when there are many senders and receivers since most if not all pairs are able to establish a connection. The extra waiting time before transmission allows for a primitive form of coordination between nodes.

5.2. RTS/CTS

We compare four sets of protocols to isolate the behavior of RTS/CTS. First, we compare PSMA, CSMA, and CSMA/CA to RTS/CTS extended versions of these protocols: PSMA/RTS/CTS, CSMA/RTS/CTS, and CSMA/CA/RTS/CTS. Then we compare PSMA/RTS/CTS/ACK, CSMA/RTS/CTS/ACK, and CSMA/CA/RTS/CTS/ACK to PSMA/ACK, CSMA/ACK, and CSMA/CA/ACK.

5.2.1 PSMA/RTS/CTS, CSMA/RTS/CTS, and CSMA/CA/RTS/CTS vs. PSMA, CSMA, and CSMA/CA.

For the variable number of hops experiments, Table 10, the addition of an RTS/CTS scheme to each of PSMA, CSMA, and CSMA/CA does not aid the protocols. It simply adds more overhead to the protocols and their throughput decreases as a result. PSMA is the least affected. The effect of the additional overhead on CSMA and CSMA/CA is most noticeable for three or more hops.

Next we examine the end-to-end vs. intermediate connections experiments. The clients cannot hear each other in the first case but can in the second. For both cases, Table 11, the incorporation of the RTS/CTS scheme into the three basic protocols does not aid any of them. PSMA/RTS/CTS is the least affected and has the highest throughput in both cases. This combination turns out to be MACA. The more conservative protocols, CSMA and CSMA/CA, pay for being a bit too cautious. In both experiments, one pair effectively captures the channel and prevents the other pair from successfully transmitting packets. RTS/CTS does not help these three in terms of fairness.

Next we consider the effect of RTS/CTS control frames when hidden terminals are present. As expected, the use of the RTS/CTS scheme greatly enhances performance of all three protocols. Table 12 shows that CSMA/RTS/CTS performs slightly better than PSMA/RTS/CTS and CSMA/CA/RTS/CTS. Note that in all cases one pair captures the channel.

When considering a situation where exposed terminals are present Table 13 shows that adding RTS/CTS control frames to CSMA greatly reduces throughput. The same table shows the throughput of PSMA and CSMA/CA is practically unchanged when RTS/CTS is added. PSMA/RTS/CTS is by far the protocol of choice for this scenario since it utilizes 97% of the channel capacity.

In the ring topology, significant gains in both throughput and fairness are achieved with the addition of RTS/CTS control frames, Table 14. Of the three basic protocols, CSMA benefits the most from the added control frames with the highest throughput per client/server pair and highest aggregate throughput. PSMA/RTS/CTS benefits the least. Performance of CSMA/CA/RTS/CTS is similar to that of CSMA/RTS/CTS. Conservative approaches pay off in this situation. Note that collision avoidance does not make much of an impact in this experiment.

When examining the two-hop cross traffic scenario, Table 15, it is apparent that the coordination function created by the RTS/CTS control frames increases the throughput and fairness of PSMA, CSMA, and CSMA/CA in the first case. In the second case only PSMA benefits from the RTS/CTS in terms of fairness and throughput. The additional overhead of the control frames reduces the performance of CSMA/RTS/CTS and CSMA/CA/RTS/CTS. The aggressive transmission policy of PSMA and the coordination afforded by RTS/CTS makes PSMA/RTS/CTS more effective than either CSMA/RTS/CTS or CSMA/CA/RTS/CTS.

Each of the protocol benefits from the addition of RTS/CTS control frames in the bottleneck experiments, Tables 16-18. All senders are able to send packets to the receivers. PSMA benefits the most from the addition of RTS/CTS than either CSMA or CSMA/CA. PSMA/RTS/CTS has a consistently higher throughput for each client server pair as well as higher aggregate throughput. The aggressive transmission policy of PSMA combined with the coordination provided by RTS/CTS does well in these experiments.

5.2.2 PSMA/RTS/CTS/ACK, CSMA/RTS/CTS/ACK and CSMA/CA/RTS/CTS/ACK vs. PSMA/ACK, CSMA/ACK and CSMA/CA/ACK

When looking at the variable hop experiments, Table 19, we see that adding RTS/CTS to PSMA/ACK and CSMA/CA/ACK reduces performance because of the increased overhead. On the other hand, CSMA performance for three or more hops benefits from the addition of the control frames. The performance of CSMA/RTS/CTS/ACK is consistently better than PSMA/RTS/CTS/ACK and CSMA/CA/RTS/CTS/ACK.

The results for the end-to-end and intermediate string experiments, Table 20, indicates that the addition of RTS/CTS gives mixed results. In the first experiment, all three protocols' performance is lessened by the addition of RTS/CTS. Note that RTS/CTS does not help the 0-5 from being locked out. The 2-3 pair captures the channel. The performance of PSMA/RTS/CTS and CSMA/RTS/CTS/ACK are comparable and both enjoy a higher aggregate throughput than CSMA/CA/RTS/CTS/ACK. In the second experiment, the aggregate throughput for PSMA/ACK and CSMA/ACK increases when the protocols are extended via RTS/CTS. Also, both pairs are able to communicate. Use of the channel is not fair by any means as the 1-4 pair still has a greater use of the channel. CSMA/CA/ACK throughput is relatively unaffected by the use of RTS/CST and capture is still a problem.

When hidden terminals are present the use of the RTS/CTS scheme greatly enhances the aggregate throughput of the three protocols, Table 21, but capture is still a problem. Both CSMA/RTS/CTS/ACK and CSMA/CA/RTS/CTS/ACK give good throughput and are preferable to PSMA/RTS/CTS/ACK. In the exposed terminal experiments, Table 22, we observe that adding control frames to PSMA/ACK increases throughput. It does so by allowing for a more orchestrated sequence of events. Capture results when RTS/CTS control frames are added to CSMA/ACK. The throughput of CSMA/CA/RTS/CTS/ACK, on the other hand, is slightly reduced. As is to be expected, the protocol with the more aggressive transmission policy performs better.

The ring experiment, Table 23, shows that PSMA/ACK and CSMA/CA/ACK both have increased throughputs when RTS/CTS control frames are added to the protocols. On the other hand, the throughput of CSMA/ACK decreases when the RTS/CTS frames are added. The throughput increase for CSMA/CA/ACK is more than a factor of three.

Table 24 shows that positive results are obtained when adding RTS/CTS control frames to the three protocols in both the cross traffic experiments. In the first case, PSMA/RTS/CTS/ACK's and CSMA/CA/RTS/CTS/ACK's performance benefits the most from RTS/CTS. Both pairs are able to communicate and the aggregate throughput is increased. In this case the performance of CSMA/RTS/CTS/ACK and CSMA/CA/RTS/CTS/ACK are comparable and both perform better than PSMA/RTS/CTS/ACK. As the number of hops increases to four, PSMA benefits the most from RTS/CTS extension. Overall, RTS/CTS increases fairness and aggregate throughput. CSMA/RTS/CTS/ACK performs the best over both experiments.

The results of the bottleneck experiments are shown in Tables 25-27. We see that each of the three protocol benefits from the addition of RTS/CTS control frames. CSMA/RTS/CTS/ACK outperforms the other two RTS/CTS extended protocols for all the bottleneck cases. In these experiments the aggressive policy of PSMA cannot compete with the more conservative approach of carrier sensing. On the other hand, a transmission policy that is too conservative does not perform well as is seen by the performance of CSMA/CA/RTS/CTS/ACK.

5.3. ACK

We compare four sets of protocols to isolate the behavior of the ACK. First, we compare PSMA, CSMA, and CSMA/CA to ACK extended versions of these protocols: PSMA/ACK, CSMA/ACK, and CSMA/CA/ACK. Then we compare PSMA/RTS/CTS/ACK, CSMA/RTS/CTS/ACK, and CSMA/CA/RTS/CTS/ACK to PSMA/RTS/CTS, CSMA/RTS/CTS, and CSMA/CA/RTS/CTS.

5.3.1 PSMA/ACK, CSMA/ACK and CSMA/CA/ACK vs. PSMA, CSMA and CSMA/CA

Table 28 shows that the contribution of acknowledgements to the performance of PSMA, CSMA, and CSMA/CA is mixed for the variable number of hops experiments. PSMA throughput benefits from the ACKs and achieves the greatest throughput. On the other hand, requiring the receipt of an ACK adversely affects the performance of both CSMA and CSMA/CA. At three hops or more the throughput of CSMA/CA/ACK is reduced to half of the non-ACK experiments where as the throughput of CSMA/ACK is reduce by one third. The poor performance of CSMA/CA is due mainly to the cumulative effect of the extra waiting time before transmitting packets, especially now that there are at least twice as many waiting periods. Acknowledgements tend to increase the throughput of a PSMA based protocol because it provides for a more orderly sequence of events rather than each node constantly sending packets. A CSMA based protocol does not gain by making use of ACKs. The primitive form of coordination that is created by the collision detection mechanism is more effective than simple ACKs. Thus combining a collision avoidance mechanism, with its extra waiting time before transmission, and acknowledgements reduces throughput. The addition of acknowledgements does more for throughput than an RTS/CTS scheme in this topology.

We next consider the string experiments with two client/server pairs, Table 29. In the case when the clients cannot hear each other, the addition of acknowledgements adds overhead and slightly decreases the performance of the three basic protocols. The 2-3 pair captures the channel. Acknowledgements fail to help the 0-5 pair in successfully transmitting packets. In effect, this experiment is reduced to a one-hop client/server pair. In the second experiment, simply adding ACKs to each of PSMA, CSMA and CSMA/CA severely degrades throughput and does nothing in terms of fairness. Not only do ACKs add to the overhead, they also increase the possibility of collisions.

If hidden terminals are present then requiring ACKs benefits all three protocols, Table 30. CSMA/CA benefits the most with throughput increasing from 19Kbps to 342 Kbps. We observe that capture is present in this experiment and ACKs do nothing towards balancing the use of the medium between the two contending clients.

Next we consider Table 31 where exposed terminals are present. The throughput of CSMA/CA, the most conservative protocol, is practically unaffected by the use of ACKs. Adding ACKs to PSMA reduces the throughput by more than 50%. Adding ACKs to CSMA allows for both pairs to communicate although there is a lack of fairness. Note that a greater reduction results after incorporating ACKs than after adding control frames.

Looking at the results of the ring experiments in Table 32 we observe that the addition of acknowledgements to the three protocols enhances performance. All pairs are able to communicate after the addition of acknowledgements. Both aggregate throughput and the throughput of each sender-receiver pair increases. Both CSMA and CSMA/CA benefit significantly. The aggregate throughput of CSMA/ACK is greater than CSMA/CA/ACK and both are greater than the aggregate throughput of PSMA/ACK. CSMA/CA/ACK is fairer than CSMA/ACK. The apparent fairness resulting from adding ACKs in this experiment is more a result of the working of the CSMA/CA contention timers rather than the ACKs themselves. Freezing the timers instead of resetting them eventually lets all clients transmit.

Table 33 shows that in the two-hop cross traffic scenario, the use of ACKs benefits all protocols in both cases with one exception. In the second case adding ACKs to CSMA/CA reduces throughput of both sender-receiver pairs. Looking at the table it does not appear that acknowledgements contribute to fairness.

Adding acknowledgements to the basic protocols benefits all protocols when a bottleneck is present, Tables 34-36. PSMA benefits the least from acknowledgements. The main effect of adding acknowledgements to PSMA are the increased number of sender-receiver pairs that are able to establish a connection. Note that very hardly any data reaches the receiver. As mentioned before CSMA/CA performs better when it has some form of feedback and this is seen in the table. CSMA benefits the most from the addition of acknowledgements. Every sender-receiver pair has non-negligible throughput. In the bottleneck experiments acknowledgements appear to contribute to fairness.

5.3.2 PSMA/RTS/CTS/ACK, CSMA/RTS/CTS/ACK and CSMA/CA/RTS/CTS/ACK vs. PSMA/RTS/CTS, CSMA/RTS/CTS, and CSMA/CA/RTS/CTS

A comparison of PSMA/RTS/CTS, CSMA/RTS/CTS, and CSMA/CA/RTS/CTS with ACK extensions of these protocols, Table 37, shows that adding ACKs has mixed results. PSMA/RTS/CTS's performance is similar in both experiments. The addition of ACKs to CSMA/RTS/CTS increases the protocol's throughput when transmitting to a destination three or more hops away. CSMA/CA/RTS/CTS is unaffected for the cases when sender and receiver are less than three hops away. Once again, the extra waiting time of CSMA/CA/RTS/CTS takes its toll when the client and server are three or more hops away.

In the first of the two string experiments (end-to-end vs. intermediate connections), Table 38, the addition of acknowledgements adds overhead and slightly decreases the performance of PSMA/RTS/CTS and CSMA/CA/RTS/CTS. On the other hand, the throughput of CSMA/RTS/CTS increases by 22% when combined with ACKs because it allows for additional coordination between sender and receiver. ACKs do nothing to alleviate capture. In the second experiment incorporating ACKs into PSMA/RTS/CTS and CSMA/RTS/CTS schemes allow for an increase in aggregate throughput. CSMA/CA/RTS/CTS throughput suffers from the added acknowledgement requirement. Again, ACKs do not help with capture.

If hidden terminals are present then adding ACKs to PSMA/RTS/CTS, CSMA/RTS/CTS, and CSMA/CA/RTS/CTS slightly decreases throughput of the first two protocols as shown in Table 39. The performance of CSMA/CA/RTS/CTS is practically unaffected. In the second of the cross experiments both PSMA/RTS/CTS and CSMA/RTS/CTS have a greater throughput after ACKs are added. The addition of ACKs retards the performance of CSMA/CA/RTS/CTS. Note that one pair captures the channel in all cases. Acknowledgements help the more aggressive protocols when then number of hops between sender and receiver is 3 or more.

If exposed terminals are present, Table 40, then the addition of ACK to PSMA/RTS/CTS, CSMA/RTS/CTS, and the basic CSMA/CA/RTS/CTS/ACK configuration either degrades throughput or has little effect on it. In the case of CSMA/RTS/CTS, adding acknowledgements results in one pair capturing the channel. Acknowledgements do nothing to increase the coordination of neighboring nodes. Acknowledgements add nothing but overhead in this situation.

Looking at the results of the ring experiment, Table 41, we see that adding ACKs to the three protocols does not have any benefit. In fact, requiring the receipt of an ACK for each transmitted frame significantly reduces the performance of PSMA/RTS/CTS and CSMA/RTS/CTS. Again, the ACKs only serve to increase the possibility of collision. The performance of CSMA/CA/RTS/CTS is practically unaffected by the incorporation of acknowledgements. Overall, acknowledgements are not of any significant benefit to protocol performance in this topology with many senders and receivers. The main benefits of increased throughput and fairness are due to the RTS/CTS control frames as mentioned above.

The results of adding ACKs to PSMA/RTS/CTS, CSMA/RTS/CTS, and CSMA/CA/RTS/CTS in the cross traffic experiments, Table 42, are mixed. The performance of PSMA/RTS/CTS is reduced. CSMA/CA/RTS/CTS throughput increases very slightly, but the throughput of CSMA/RTS/CTS shows greater increase when ACKs are employed. CSMA/RTS/CTS benefits from the extra coordination between sender and receiver. When the number of hops of between each pair increases to four, ACKs help all protocols except CSMA/CA/RTS/CTS. CSMA/RTS/CTS/ACK performs better overall in these experiments.

In the zero-hop bottleneck experiment, Table 43, the incorporation of ACKs into PSMA/RTS/CTS decreases throughput. CSMA/RTS/CTS with ACKs has more than twice the throughput than without ACKs. CSMA/CA/RTS/CTS is relatively unchanged. The trend continues as the length of the bottleneck increases Tables 44 and 45. Only CSMA/RTS/CTS benefits from the use of ACKs and the throughput of each pair doubles. The added coordination between the nodes is enhanced greatly in this case. The ACKs do nothing in terms of fairness.

6. Conclusion

Acknowledgements help improve the throughput of the basic protocols in the hidden terminal, ring, cross, and bottleneck experiments. The increase in coordination between sender-receiver is the main result. Acknowledgements tends to aid CSMA performance more than it aids the performance of PSMA and CSMA/CA. The acknowledgements add overhead in the variable hops, end-to-end and intermediate, and exposed terminal experiments. When RTS/CTS control frames are already being used the addition of acknowledgements tends to add to the overhead and degrades performance. This is due to the increased possibility of collisions. PSMA/RTS/CTS throughput usually degrades with the addition of ACKs. CSMA/CA/RTS/CTS performance is either unaffected or it deteriorates depending on the particular experiment. Only CSMA/RTS/CTS tends to benefit from the incorporation of acknowledgements with the exception of the ring and hidden terminal experiments. There is no evidence that acknowledgements contribute to the fairness of the various protocols.

The use of RTS/CTS control frames aids the three basic protocols in all but the variable hop, end-to-end and intermediate hop, and two of the exposed terminal experiments. Adding RTS/CTS to CSMA increases performance in the exposed terminal experiment. Of the three basic protocols PSMA benefits the most overall from the use of RTS/CTS. The incorporation of RTS/CTS control frames to protocols making use of acknowledgements more often than not leads to increased throughput. RTS/CTS is responsible for the fairness (i.e. node coordination) of the protocols that employ it. By being fair and allowing all clients to successfully transmit packets it increases the throughput of each client/server pair. In summary, the degree of coordination is greater when using RTS/CTS than when using acknowledgements.

When communication using unreliable transmission it is desirable to be as aggressive as possible and have some form of coordination between nodes. Of the three basic protocols (PSMA, CSMA, and CSMA/CA), two variations emerge as the most robust PSMA/RTS/CTS and CSMA/RTS/CTS/ACK. Not surprisingly PSMA/RTS/CTS is better know as MACA. CSMA/RTS/CTS/ACK has a slight advantage in the variable hop experiments. In the end-to-end and intermediate experiments, MACA is a bit better when the senders know of each other otherwise CSMA/RTS/CTS/ACK has a small advantage. MACA is by far the better of the two in the ring experiment. The is practically no difference in performance in the hidden terminal experiment. MACA fares better when exposed terminals are present. The first of the two cross experiments shows no advantage to either protocol but CSMA/RTS/CTS/ACK's throughput is three times greater than MACA's when the number of hops increases. CSMA/RTS/CTS/ACT has a slight advantage in the bottleneck experiments. When considering the different experiments MACA is a bit more robust.

Two other important components should also be considered. Specifically, the backoff schemes and the yield time used in MAC protocols. Yield time is defined as the amount of time the sender backs off before sending another data frame after transmitting a data frame. Yield time also refers to the amount of time the receiver backs off before sending a frame after receiving a data frame. By adjusting the backoff schemes and yield time parameters, we can conceivably manipulate the outcome of the network performance. For example, if we implemented a less aggressive backoff timer or use a longer yield time, fairness can be achieved. Therefore, further research is needed regarding the MAC layer timers.

Reference

[Bagrodia98] R. Bagrodia, R. Meyerr, et al., “PARSEC: A Parallel Simulation Environment for Complex System”, Computer Magazine, 1998.

[Bharghavan94] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LAN's," ACM SIGCOMM, 1994.

[Editors96] Editors of IEEE 802.11, Wireless LAN Medium Access Control (MAC and Physical Layer (PHY) specifications, Draft Standard IEEE 802.11, P802.11/D5.0, July 1996.

[Fullmer95] C. Fullmer and J.J. Garcia-Luna-Aceves, “Floor Acquisition Multiple Access (FAMA) for packet radio networks”, Computer Communication Review, vol. 25, (no. 4), (ACM SIGCOMM '95, Cambridge, MA, USA, 28 Aug.-1 Sept. 1995.) ACM, Oct. 1995.

[Gerla99] Mario Gerla, Ken Tang, Rajive Bagrodia, “TCP Performance in Wireless Multihop Networks”, Proceedings of IEEE WMCSA'99, Feb. 1999.

[Haartsen98] J. Haartsen, M. Naghshineh, J. Inouye, O.J. Joeressen, and W. Allen, "Bluetooth: Vision, Goals, and Architecture", ACM SIGMOBILE Mobile Computing and Communications Review, vol. 2, no. 4, Oct. 1998, pp. 38-45.

[Joseph98] Anthony Joseph, B. R. Badrinath, and Randy Katz, A Case for Services over Cascaded Networks. First ACM/IEEE International Conference on Wireless and Mobile Multimedia (WoWMoM'98), October 30, 1998, Dallas Texas.

[Jubin87] John Jubin and Janet D. Tornow, “The DARPA Packet Radio Network Protocols”, Proceedings of the IEEE, Jan. 1987.

[Karn90] P. Karn, “MACA –a New Channel Access Method for Packet Radio”, in ARRL/CRRL Amateur radio 9th Computer Networking Conference, ARRL, 1990.

[Negus98] K.J. Negus, J. Waters, J. Tourrilhes, C. Romans, J. Lansford, and S. Hui, "HomeRF and SWAP: Wireless Networking for the Connected Home", ACM SIGMOBILE Mobile Computing and Communications Review, vol. 2, no. 4, Oct. 1998, pp. 28-37.

[Tanenbaum96] Andrew S. Tanenbaum, “Computer Networks: Third Edition”, Prentice Hall PTR, New Jersey, 1996.

[Zeng98] X. Zeng, R. Bagrodia and M. Gerla, “GloMoSim: a Library for the Parallel Simulation of Large-scale Wireless Networks”, PADS, 1998.

Appendix

Table 1.
Throughput (Kbps), Single Client/Server Pair.
Variable Number of Hops

Number of Hops	PSMA	CSMA	CSMA/CA
1	1941.5	1946.7	1834.8
2	647.2	964.0	836.2
3	485.4	644.0	608.7
4	485.4	644.0	601.5
5	485.4	644.0	601.4

Table 2.
Throughput (Kbps), Two Client/Server Pairs.
Intermediate and End-to-end

Client/Server	PSMA	CSMA	CSMA/CA
0-5	0 (NE)	0 (NE)	0 (NE)
2-3	1941.5	1946.7	1834.9

0-5	0 (NE)	0 (NE)	0 (NE)
1-4	485.4	646.8	570.8

Table 3.
Throughput (Kbps), Hidden Terminal

Client/Server	PSMA	CSMA	CSMA/CA
0-1	0 (NE)	0 (NE)	0 (NE)
2-1	20.0	20.0	18.9

Table 4.
Throughput (Kbps), Exposed Terminal

Client/Server	PSMA	CSMA	CSMA/CA
1-0	1941.5	0 (NE)	967.8
2-3	1941.5	1946.7	967.1

Table 5.
Throughput (Kbps) in Ring Topology

Client/Server	PSMA	CSMA	CSMA/CA
0-1	0 (NE)	0 (NE)	0.1
1-2	0 (NE)	0 (NE)	0.2
2-3	0 (NE)	0 (NE)	1.6
3-4	0 (NE)	0 (NE)	0.4
4-5	0 (NE)	0 (NE)	0.1
5-6	0 (NE)	0 (NE)	0.2
6-7	0 (NE)	0 (NE)	6.6
7-0	58.4	58.5	49.9

Table 6.
Throughput (Kbps), Cross.

Client/Server	PSMA	CSMA	CSMA/CA
1-3	0 (NE)	0 (NE)	0 (NE)
6-7	16.3	1.5	22.8

0-4	0 (NE)	0.2	31.8
5-8	12.3	16.5	48.2

Table 7.
Throughput (Kbps), Bottleneck.

Client/Server	PSMA	CSMA	CSMA/CA
0-5	0 (NE)	0.6	1.2
1-6	0 (NE)	1.3	1.7
2-7	0 (NE)	1.7	2.2
3-8	0 (NE)	4.8	3.8
4-9	3.3	8.8	9.0

Table 8.
Throughput (Kbps), One Hop Bottleneck.

Client/Server	PSMA	CSMA	CSMA/CA
0-5	0 (NE)	0 (NE)	0 (NE)
1-6	0 (NE)	0 (NE)	0.3
2-7	0 (NE)	0 (NE)	0.2
3-8	0 (NE)	0.8	1.8
4-9	2.5	4.2	4.8

Table 9.
Throughput (Kbps), Two Hop Bottleneck.

Client/Server	PSMA	CSMA	CSMA/CA
0-5	0 (NE)	0 (NE)	0.4
1-6	0 (NE)	0 (NE)	0.1
2-7	0 (NE)	0 (NE)	0.2
3-8	0 (NE)	0.8	0.9
4-9	2.5	4.2	3.3

Table 10.
Throughput (Kbps), Single Client/Server Pair.
Variable Number of Hops

Hops	PSMA	CSMA	CSMA/CA	PSMA/RTS/CTS	CSMA/RTS/CTS	CSMA/CA/RTS/CTS
1	1941.5	1946.7	1834.8	1901.1	1807.4	1788.8
2	647.2	964.0	836.2	726.0	912.6	820.0
3	485.4	644.0	608.7	468.4	455.8	547.5
4	485.4	644.0	601.5	466.1	442.7	431.0
5	485.4	644.0	601.4	466.1	389.1	399.9

Table 11.
Throughput (Kbps), Two Client/Server Pairs.
Intermediate and End-to-end

Client/Server	PSMA	CSMA	CSMA/CA	PSMA/RTS/CTS	CSMA/RTS/CTS	CSMA/CA/RTS/CTS
0-5	0 (NE)	0 (NE)	0 (NE)	0 (NE)	0 (NE)	0 (NE)
2-3	1941.5	1946.7	1834.9	1901.1	1532.2	1527.8
0-5	0 (NE)	0 (NE)	0 (NE)	0 (NE)	0 (NE)	0.3
1-4	485.4	646.8	570.8	471.6	443.8	250.4

Table 12.
Throughput (Kbps), Hidden Terminal

Client/Server	PSMA	CSMA	CSMA/CA	PSMA/RTS/CTS	CSMA/RTS/CTS	CSMA/CA/RTS/CTS
0-1	0 (NE)	0 (NE)	0 (NE)	0 (NE)	0 (NE)	0 (NE)
2-1	20.0	20.0	18.9	1712.7	1811.9	1755.3

Table 13.
Throughput (Kbps), Exposed Terminal

Client/Server	PSMA	CSMA	CSMA/CA	PSMA/RTS/CTS	CSMA/RTS/CTS	CSMA/CA/RTS/CTS
1-0	1941.5	0 (NE)	967.8	1901.1	919.7	915.7
2-3	1941.5	1946.7	967.1	1901.1	922.3	920.7

Table 14.
Throughput (Kbps) in Ring Topology

Client/Server	PSMA	CSMA	CSMA/CA	PSMA/RTS/CTS	CSMA/RTS/CTS	CSMA/CA/RTS/CTS
0-1	0 (NE)	0 (NE)	0.1	402.1	432.4	442.3
1-2	0 (NE)	0 (NE)	0.2	392.5	439.0	431.2
2-3	0 (NE)	0 (NE)	1.6	398.6	433.2	403.6
3-4	0 (NE)	0 (NE)	0.4	391.6	441.7	419.1
4-5	0 (NE)	0 (NE)	0.1	391.4	442.1	436.6
5-6	0 (NE)	0 (NE)	0.2	389.8	430.4	411.1
6-7	0 (NE)	0 (NE)	6.6	399.2	432.4	394.0
7-0	58.4	58.5	49.9	436.7	472.0	419.1

Table 15.
Throughput (Kbps), Cross.

Client/Server	PSMA	CSMA	CSMA/CA	PSMA/RTS/CTS	CSMA/RTS/CTS	CSMA/CA/RTS/CTS
1-3	0 (NE)	0 (NE)	0 (NE)	428.0	315.7	404.2
6-7	16.3	1.5	22.8	437.1	321.9	384.0
0-4	0 (NE)	0.2	31.8	58.1	5.2	26.3
5-8	12.3	16.5	48.2	66.0	17.9	37.4

Table 16.
Throughput (Kbps), Bottleneck.

Client/Server	PSMA	CSMA	CSMA/CA	PSMA/RTS/CTS	CSMA/RTS/CTS	CSMA/CA/RTS/CTS
0-5	0 (NE)	0.6	1.2	168.8	63.0	41.5
1-6	0 (NE)	1.3	1.7	131.9	62.8	41.3
2-7	0 (NE)	1.7	2.2	140.3	61.6	43.3
3-8	0 (NE)	4.8	3.8	134.6	65.7	45.0
4-9	3.3	8.8	9.0	172.6	70.1	47.7

Table 17.
Throughput (Kbps), One Hop Bottleneck.

Client/Server	PSMA	CSMA	CSMA/CA	PSMA/RTS/CTS	CSMA/RTS/CTS	CSMA/CA/RTS/CTS
0-5	0 (NE)	0 (NE)	0 (NE)	105.2	44.7	75.8
1-6	0 (NE)	0 (NE)	0.3	109.8	43.6	69.5
2-7	0 (NE)	0 (NE)	0.2	105..4	47.0	68.5
3-8	0 (NE)	0.8	1.8	133.8	46.8	74.1
4-9	2.5	4.2	4.8	108.7	50.9	70.6

Table 18.
Throughput (Kbps), Two Hop Bottleneck.

Client/Server	PSMA	CSMA	CSMA/CA	PSMA/RTS/CTS	CSMA/RTS/CTS	CSMA/CA/RTS/CTS
0-5	0 (NE)	0 (NE)	0.4	97.1	42.6	5.2
1-6	0 (NE)	0 (NE)	0.1	100.0	44.5	3.9
2-7	0 (NE)	0 (NE)	0.2	94.8	48.4	4.4
3-8	0 (NE)	0.8	0.9	98.3	46.4	4.7
4-9	2.5	4.2	3.3	96.6	50.2	5.1

Table 19.
Throughput (Kbps), Single Client/Server Pair.
Variable Number of Hops

Hops	PSMA/ACK	CSMA/ACK	CSMA/CA/ACK	PSMA/RTS/CTS/ACK	CSMA/RTS/CTS/ACK	CSMA/CA/RTS/CTS/ACK
1	1911.0	1911.0	1791.4	1862.2	1862.3	1747.4
2	931.8	705.3	827.2	713.1	931.1	805.7
3	606.0	420.2	317.5	464.5	508.6	486.5
4	606.0	411.8	177.0	463.1	508.6	137.0
5	606.0	404.7	166.0	461.6	508.6	131.5

Table 20.
Throughput (Kbps), Two Client/Server Pairs.
Intermediate and End-to-end

Client/Server	PSMA/ACK	CSMA/ACK	CSMA/CA/ACK	PSMA/RTS/CTS/ACK	CSMA/RTS/CTS/ACK	CSMA/CA/RTS/CTS/ACK
0-5	0 (NE)	0 (NE)	0 (NE)	0 (NE)	0 (NE)	0 (NE)
2-3	1911.0	1911.0	1791.5	1862.2	1862.2	1512.3
0-5	0 (NE)	0 (NE)	0 (NE)	0 (NE)	0 (NE)	0 (NE)
1-4	140.0	284.5	152.4	584.8	534.3	146.7

Table 21.
Throughput (Kbps), Hidden Terminal

Client/Server	PSMA/ACK	CSMA/ACK	CSMA/CA/ACK	PSMA/RTS/CTS/ACK	CSMA/RTS/CTS/ACK	CSMA/CA/RTS/CTS/ACK
0-1	0 (NE)	0 (NE)	0 (NE)	0 (NE)	0 (NE)	0 (NE)
2-1	278.1	283.5	342.0	1636.0	1737.8	1714.3

Table 22.
Throughput (Kbps), Exposed Terminal

Client/Server	PSMA/ACK	CSMA/ACK	CSMA/CA/ACK	PSMA/RTS/CTS/ACK	CSMA/RTS/CTS/ACK	CSMA/CA/RTS/CTS/ACK
1-0	614.7	809.5	916.2	913.4	0 (NE)	892.9
2-3	627.8	1082.3	923.3	916.0	1862.2	900.6

Table 23.
Throughput (Kbps) in Ring Topology

Client/Server	PSMA/ACK	CSMA/ACK	CSMA/CA/ACK	PSMA/RTS/CTS/ACK	CSMA/RTS/CTS/ACK	CSMA/CA/RTS/CTS/ACK
0-1	9.4	96.6	86.7	49.4	30.2	433.1
1-2	7.5	98.2	84.7	54.0	33.5	420.9
2-3	8.9	96.8	86.7	50.0	33.0	391.8
3-4	7.4	92.7	89.6	46.5	35.1	400.2
4-5	7.7	98.3	88.5	53.1	47.5	444.9
5-6	7.8	96.8	86.4	51.1	34.8	412.1
6-7	9.8	98.0	100.2	57.3	37.0	392.7
7-0	33.3	135.8	102.5	92.6	87.8	446.6

Table 24.
Throughput (Kbps), Cross.

Client/Server	PSMA/ACK	CSMA/ACK	CSMA/CA/ACK	PSMA/RTS/CTS/ACK	CSMA/RTS/CTS/ACK	CSMA/CA/RTS/CTS/ACK
1-3	65.5	173.2	131.1	364.4	440.2	421.2
6-7	85.2	180.0	155.9	376.8	449.9	417.6
0-4	37.6	67.7	4.0	195.2	195.6	10.5
5-8	49.4	77.1	9.2	210.1	198.3	14.9

Table 25.
Throughput (Kbps), Bottleneck.

Client/Server	PSMA/ACK	CSMA/ACK	CSMA/CA/ACK	PSMA/RTS/CTS/ACK	CSMA/RTS/CTS/ACK	CSMA/CA/RTS/CTS/ACK
0-5	0.1	135.2	35.9	63.1	180.8	41.4
1-6	0.1	149.9	38.0	63.8	180.1	40.6
2-7	0.2	147.1	38.6	62.2	180.5	42.5
3-8	0.7	149.4	35.9	68.5	181.5	41.4
4-9	5.4	152.3	42.7	71.0	185.7	46.8

Table 26.
Throughput (Kbps), One Hop Bottleneck.

Client/Server	PSMA/ACK	CSMA/ACK	CSMA/CA/ACK	PSMA/RTS/CTS/ACK	CSMA/RTS/CTS/ACK	CSMA/CA/RTS/CTS/ACK
0-5	0.9	92.6	5.1	84.9	123.9	19.0
1-6	0 (NE)	92.4	6.2	90.9	121.4	18.3
2-7	0 (NE)	91.9	5.5	85.7	101.2	21.2
3-8	0.2	93.5	5.7	90.6	121.7	23.8
4-9	3.5	101.5	9.6	94.8	123.1	24.0

Table 27.
Throughput (Kbps), Two Hop Bottleneck.

Client/Server	PSMA/ACK	CSMA/ACK	CSMA/CA/ACK	PSMA/RTS/CTS/ACK	CSMA/RTS/CTS/ACK	CSMA/CA/RTS/CTS/ACK
0-5	0 (NE)	86.5	4.3	53.0	123.9	3.4
1-6	0 (NE)	87.3	3.0	53.4	121.4	3.5
2-7	0.1	87.8	4.0	51.2	101.2	2.8
3-8	0.2	85.1	3.4	52.6	121.7	2.2
4-9	3.3	91.4	5.7	55.4	123.1	7.6

Table 28.
Throughput (Kbps), Single Client/Server Pair.
Variable Number of Hops

Hops	PSMA	CSMA	CSMA/CA	PSMA/ACK	CSMA/ACK	CSMA/CA/ACK
1	1941.5	1946.7	1834.8	1911.0	1911.0	1791.4
2	647.2	964.0	836.2	931.8	705.3	827.2
3	485.4	644.0	608.7	606.0	420.2	317.5
4	485.4	644.0	601.5	606.0	411.8	177.0
5	485.4	644.0	601.4	606.0	404.7	166.0

Table 29.
Throughput (Kbps), Two Client/Server Pairs.
Intermediate and End-to-end

Client/Server	PSMA	CSMA	CSMA/CA	PSMA/ACK	CSMA/ACK	CSMA/CA/ACK
0-5	0 (NE)	0 (NE)	0 (NE)	0 (NE)	0 (NE)	0 (NE)
2-3	1941.5	1946.7	1834.9	1911.0	1911.0	1791.5
0-5	0 (NE)	0 (NE)	0 (NE)	0 (NE)	0 (NE)	0 (NE)
1-4	485.4	646.8	570.8	140.0	284.5	152.4

Table 30.
Throughput (Kbps), Hidden Terminal

Client/Server	PSMA	CSMA	CSMA/CA	PSMA/ACK	CSMA/ACK	CSMA/CA/ACK
0-1	0 (NE)	0 (NE)	0 (NE)	0 (NE)	0 (NE)	0 (NE)
2-1	20.0	20.0	18.9	278.1	283.5	342.0

Table 31.
Throughput (Kbps), Exposed Terminal

Client/Server	PSMA	CSMA	CSMA/CA	PSMA/ACK	CSMA/ACK	CSMA/CA/ACK
1-0	1941.5	0 (NE)	967.8	614.7	809.5	916.2
2-3	1941.5	1946.7	967.1	627.8	1082.3	923.3

Table 32.
Throughput (Kbps) in Ring Topology

Client/Server	PSMA	CSMA	CSMA/CA	PSMA/ACK	CSMA/ACK	CSMA/CA/ACK
0-1	0 (NE)	0 (NE)	0.1	9.4	96.6	86.7
1-2	0 (NE)	0 (NE)	0.2	7.5	98.2	84.7
2-3	0 (NE)	0 (NE)	1.6	8.9	96.8	86.7
3-4	0 (NE)	0 (NE)	0.4	7.4	92.7	89.6
4-5	0 (NE)	0 (NE)	0.1	7.7	98.3	88.5
5-6	0 (NE)	0 (NE)	0.2	7.8	96.8	86.4
6-7	0 (NE)	0 (NE)	6.6	9.8	98.0	100.2
7-0	58.4	58.5	49.9	33.3	135.8	102.5

Table 33.
Throughput (Kbps), Cross.

Client/Server	PSMA	CSMA	CSMA/CA	PSMA/ACK	CSMA/ACK	CSMA/CA/ACK
1-3	0 (NE)	0 (NE)	0 (NE)	65.5	173.2	131.1
6-7	16.3	1.5	22.8	85.2	180.0	155.9
0-4	0 (NE)	0.2	31.8	37.6	67.7	4.0
5-8	12.3	16.5	48.2	49.4	77.1	9.2

Table 34.
Throughput (Kbps), Bottleneck.

Client/Server	PSMA	CSMA	CSMA/CA	PSMA/ACK	CSMA/ACK	CSMA/CA/ACK
0-5	0 (NE)	0.6	1.2	0.1	135.2	35.9
1-6	0 (NE)	1.3	1.7	0.1	149.9	38.0
2-7	0 (NE)	1.7	2.2	0.2	147.1	38.6
3-8	0 (NE)	4.8	3.8	0.7	149.4	35.9
4-9	3.3	8.8	9.0	5.4	152.3	42.7

Table 35.
Throughput (Kbps), One Hop Bottleneck.

Client/Server	PSMA	CSMA	CSMA/CA	PSMA/ACK	CSMA/ACK	CSMA/CA/ACK
0-5	0 (NE)	0 (NE)	0 (NE)	0.9	92.6	5.1
1-6	0 (NE)	0 (NE)	0.3	0 (NE)	92.4	6.2
2-7	0 (NE)	0 (NE)	0.2	0 (NE)	91.9	5.5
3-8	0 (NE)	0.8	1.8	0.2	93.5	5.7
4-9	2.5	4.2	4.8	3.5	101.5	9.6

Table 36.
Throughput (Kbps), Two Hop Bottleneck.

Client/Server	PSMA	CSMA	CSMA/CA	PSMA/ACK	CSMA/ACK	CSMA/CA/ACK
0-5	0 (NE)	0 (NE)	0.4	0 (NE)	86.5	4.3
1-6	0 (NE)	0 (NE)	0.1	0 (NE)	87.3	3.0
2-7	0 (NE)	0 (NE)	0.2	0.1	87.8	4.0
3-8	0 (NE)	0.8	0.9	0.2	85.1	3.4
4-9	2.5	4.2	3.3	3.3	91.4	5.7

Table 37.
Throughput (Kbps), Single Client/Server Pair.
Variable Number of Hops

Hops	MA/RTS/C	SMA/RTS/CT	MA/CA/RTS/C	PSMA/RTS/CTS/ACK	CSMA/RTS/CTS/ACK	CSMA/CA/RTS/CTS/ACK
1	1901.1	1807.4	1788.8	1862.2	1862.3	1747.4
2	726.0	912.6	820.0	713.1	931.1	805.7
3	468.4	455.8	547.5	464.5	508.6	486.5
4	466.1	442.7	431.0	463.1	508.6	137.0
5	466.1	389.1	399.9	461.6	508.6	131.5

Table 38.
Throughput (Kbps), Two Client/Server Pairs.
Intermediate and End-to-end

Client/Server	MA/RTS/C	SMA/RTS/CT	MA/CA/RTS/C	PSMA/RTS/CTS/ACK	CSMA/RTS/CTS/ACK	CSMA/CA/RTS/CTS/ACK
0-5	0 (NE)	0 (NE)	0 (NE)	0 (NE)	0 (NE)	0 (NE)
2-3	1901.1	1532.2	1527.8	1862.2	1862.2	1512.3

0-5	0 (NE)	0 (NE)	0.3	0 (NE)	0 (NE)	0 (NE)
1-4	471.6	443.8	250.4	584.8	534.3	146.7

Table 39.
Throughput (Kbps), Hidden Terminal

Client/Server	MA/RTS/C	SMA/RTS/CT	SMA/CA/RTS/C	PSMA/RTS/CTS/ACK	CSMA/RTS/CTS/ACK	CSMA/CA/RTS/CTS/ACK
0-1	0 (NE)	0 (NE)	0 (NE)	0 (NE)	0 (NE)	0 (NE)
2-1	1712.7	1811.9	1755.3	1636.0	1737.8	1714.3

Table 40.
Throughput (Kbps), Exposed Terminal

Client/Server	MA/RTS/C	SMA/RTS/CT	SMA/CA/RTS/C	PSMA/RTS/CTS/ACK	CSMA/RTS/CTS/ACK	CSMA/CA/RTS/CTS/ACK
1-0	1901.1	919.7	915.7	913.4	0 (NE)	892.9
2-3	1901.1	922.3	920.7	916.0	1862.2	900.6

Table 41.
Throughput (Kbps) in Ring Topology

Client/Server	MA/RTS/C	SMA/RTS/CT	SMA/CA/RTS/C	PSMA/RTS/CTS/ACK	CSMA/RTS/CTS/ACK	CSMA/CA/RTS/CTS/ACK
0-1	402.1	432.4	442.3	49.4	30.2	433.1
1-2	392.5	439.0	431.2	54.0	33.5	420.9
2-3	398.6	433.2	403.6	50.0	33.0	391.8
3-4	391.6	441.7	419.1	46.5	35.1	400.2
4-5	391.4	442.1	436.6	53.1	47.5	444.9
5-6	389.8	430.4	411.1	51.1	34.8	412.1
6-7	399.2	432.4	394.0	57.3	37.0	392.7
7-0	436.7	472.0	419.1	92.6	87.8	446.6

Table 42.
Throughput (Kbps), Cross.

Client/Server	MA/RTS/C	SMA/RTS/CT	SMA/CA/RTS/C	PSMA/RTS/CTS/ACK	CSMA/RTS/CTS/ACK	CSMA/CA/RTS/CTS/ACK
1-3	428.0	315.7	404.2	364.4	440.2	421.2
6-7	437.1	321.9	384.0	376.8	449.9	417.6
0-4	58.1	5.2	26.3	195.2	195.6	10.5
5-8	66.0	17.9	37.4	210.1	198.3	14.9

Table 43.
Throughput (Kbps), Bottleneck.

Client/Server	MA/RTS/C	SMA/RTS/CT	SMA/CA/RTS/C	PSMA/RTS/CTS/ACK	CSMA/RTS/CTS/ACK	CSMA/CA/RTS/CTS/ACK
0-5	168.8	63.0	41.5	63.1	180.8	41.4
1-6	131.9	62.8	41.3	63.8	180.1	40.6
2-7	140.3	61.6	43.3	62.2	180.5	42.5
3-8	134.6	65.7	45.0	68.5	181.5	41.4
4-9	172.6	70.1	47.7	71.0	185.7	46.8

Table 44.
Throughput (Kbps), One Hop Bottleneck.

Client/Server	MA/RTS/C	SMA/RTS/CT	SMA/CA/RTS/C	PSMA/RTS/CTS/ACK	CSMA/RTS/CTS/ACK	CSMA/CA/RTS/CTS/ACK
0-5	105.2	44.7	75.8	84.9	123.9	19.0
1-6	109.8	43.6	69.5	90.9	121.4	18.3
2-7	105..4	47.0	68.5	85.7	101.2	21.2
3-8	133.8	46.8	74.1	90.6	121.7	23.8
4-9	108.7	50.9	70.6	94.8	123.1	24.0

Table 45.
Throughput (Kbps), Two Hop Bottleneck.

Client/Server	MA/RTS/C	SMA/RTS/CT	SMA/CA/RTS/C	PSMA/RTS/CTS/ACK	CSMA/RTS/CTS/ACK	CSMA/CA/RTS/CTS/ACK
0-5	97.1	42.6	5.2	53.0	123.9	3.4
1-6	100.0	44.5	3.9	53.4	121.4	3.5
2-7	94.8	48.4	4.4	51.2	101.2	2.8
3-8	98.3	46.4	4.7	52.6	121.7	2.2
4-9	96.6	50.2	5.1	55.4	123.1	7.6