Isolation of Wireless Ad hoc
Medium Access Mechanisms under TCP

Ken Tang, Mario Correa, Mario Gerla
Computer Science Department
University of California, Los Angeles
{ktang, mcorrea, 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 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

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. We isolate individual medium access mechanisms (such as carrier sensing, RTS/CTS control frames and link-level ACKs) through experiments to determine the effect each mechanism has on the network.

Section 2 discusses our research study. Simulation experiments are described in section 3. Results from our TCP experiments under various MAC protocols are presented in section 4. Finally, section 5 concludes the paper. All figures are shown in the Appendix.

2. RESEARCH GOALS

Existing random access, wireless MAC protocols include Carrier Sense Multiple Access (CSMA), Multiple Access with Collision Avoidance (MACA) [7], MACAW [2], Floor Acquisition Multiple Access (FAMA) [5] and IEEE 802.11 [4]. These various random access schemes for wireless ad hoc networks employ different methods, or mechanisms, for channel contention on the wireless medium. Among them are carrier sense, packet sense (no carrier sensing), 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.

In our study, we isolate each of the above features to determine the individual effect each mechanism incurs on network performance and at the same time determine which combination works best under TCP. In particular, we hope to answer the following questions: First of all, what kind of basic medium access technique works best? Is it PSMA, CSMA or CSMA/CA? Also, it is well known that RTS/CTS improves the system performance of wireless networks. However, does the improvements aid in fairness, aggregate throughput or both? Furthermore, link-level ACKs have been suggested to alleviate the problems associated with the high error rates of the wireless channel. To what extent are link-level ACKs useful? Finally, with many possible combinations of wireless access mechanisms, which combination works best? Although there are many important metrics to consider, in this paper we focus on throughput and fairness. For our study, we implemented various combinations of the above features and compare them to one another in an effort to isolate each mechanism. The protocols implemented are packet sense (PSMA), carrier sense (CSMA), carrier sense with collision avoidance (CSMA/CA), PSMA/RTS/CTS, CSMA/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 [9]. GloMoSim is a discrete event parallel simulation environment implemented in PARSEC, PARallel Simulation Environment for Complex Systems [1]. It includes various detailed, 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.

3. EXPERIMENTAL CONFIGURATIONS AND PARAMETERS

For our simulation experiments, we consider several topologies (Fig. 1 - Fig. 4): string, hidden terminal, exposed terminal, and ring. The arrows represent the direction of data packet transmissions. FTP with infinite backlog running on top of TCP is used for the application. We
utilize static routing to route packets since mobility is not considered. 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, CSMA/CA/RTS/CTS/ACK. Each MAC protocol is uses exponential backoff. 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). No capture ability of the radios and free-space channel model are used for easier understanding of the results due to their deterministic behavior as opposed to radios with capture ability and fading channel model, where there are some randomness involved. Transmission and propagation delays are modeled. Processing delay is negligible.

---

Figure 1. String Topology.

Figure 2. Hidden Terminal Topology.

Figure 3. Exposed Terminal Topology.

Figure 4. Ring Topology.

Each node has a 25-packet MAC layer buffer pool. Scheduling of packet transmissions is FIFO. The TCP simulation code was ported directly from FreeBSD 2.2.4 code (TCP-Lite) into GloMoSim. TCP packet length is assumed fixed at 1460B. In our experiments, we "force" the maximum TCP advertised window to be one packet size (i.e. 1460B) in order to better understand and analyze the network behavior. Also, it is shown in [6] that having an adaptive window does not help but rather worsens TCP's performance under a wireless ad hoc environment. TCP connections are started uniformly, distributed between 0 to 10 seconds. Each simulation run is executed for 200 simulated seconds.

Four 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 Fig. 1. Each node is only within range of their intermediate neighbors. This experiment useful to determine the effectiveness of each protocol when no contention exists. Next, we divert our attention to the hidden terminal environment depicted in Fig. 2. 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 Fig. 3, 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 Fig. 4 to determine each protocol's fairness characteristics. 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.

4. FILE TRANSFER WITH TCP

To determine the affects of the individual medium access mechanisms (PSMA, CSMA, CSMA/CA, RTS/CTS and ACK) under TCP, we contrast various protocols using the experiments mentioned in section 3. The traffic type is FTP with infinite backlog at each source node.

A. PSMA, CSMA AND CSMA/CA

We start by examining the performance of TCP on top of PSMA, CSMA and CSMA/CA with the variable number of hops experiments. Here no contention exists and therefore, intuitively, we expect PSMA and CSMA to exhibit similar network behavior since carrier sensing is not needed. CSMA/CA, meanwhile, produces less throughput due to the unnecessary collision avoidance overhead. Fig. 1 confirms our insight. The throughput behavior shown in Fig. 1 also supports the send-and-wait model (since TCP window size is set to one packet size).

Now, let us refer to the hidden terminal experiments. All three protocols (PSMA, CSMA and CSMA/CA) mirror one another (Fig. 2). The connection from node 2 to node 1 captures the channel and locks out the connection from node 0 to node 1 completely. The capture behavior is explained by the interplay of MAC timeouts with TCP timeouts, coupled with the presence of undetected link losses. When two sessions compete for the same channel and one is "pushed back" by the timeouts, the binary backoff nature of both MAC and TCP timeouts makes the situation
progressively worse for the loser. Note that the collision avoidance scheme of CSMA/CA alone does not work well without feedback from control frame losses, such as RTS or ACK frames. This is because the MAC backoff value does not increase when collisions of data frames due to contention occur since the MAC layer has no way of finding out about the collisions. Therefore, the collision avoidance part of CSMA/CA has no beneficial effect. In terms of throughput, CSMA outperforms PSMA and CSMA/CA.

In the exposed terminal experiments (Fig. 3), the connection between node 1 and node 0 is totally locked out in PSMA. This is counterintuitive since we expect a MAC layer protocol that does not carrier sense to do well in the exposed terminal scenario [7]. However, since TCP acknowledges every data packet, the TCP ACK may deter node 0 from sending TCP ACK to node 1 since node 1 cannot receive any data packets from node 0 while node 2 is transmitting to node 3. What makes things worse is the fact that since the transmission delay of TCP ACKs are minimal compared to the data packets, the data packets from node 2 will more likely interfere with the ACKs sent by node 0 to node 1. This scenario causes TCP ACKs from node 0 to be lost, resulting in retransmissions by node 1. CSMA and CSMA/CA fare better in both overall throughput and fairness because these protocols carrier senses before transmitting both TCP data and TCP ACKs. A busy channel triggers exponential backoff and thus allows one of the nodes to transmit successfully.

To further analyze the effects of PSMA, CSMA and CSMA/CA, we next consider the ring experiments (Fig. 4). Under the ring experiments, PSMA achieves better fairness and aggregate throughput compare to that of CSMA and CSMA/CA. PSMA obtains an aggregate throughput of 3Mbps while CSMA and CSMA/CA both achieve 2.1Mbps. PSMA’s throughput of 3Mbps is quite good considering the fact that the maximum theoretical throughput achievable on the ring is 4Mbps (i.e., two simultaneous transmissions at least 3 hop apart). We note here again that CSMA/CA has no benefit compare to CSMA. Without control frames to determine the status of the receiver, the collision avoidance mechanism of CSMA/CA does not work well and thus behaves similar to plain CSMA. Therefore, the behaviors of both protocols are comparable throughout all experiments.

B. RTS/CTS

To isolate the impact of RTS/CTS control frames, we contrast the following protocols: PSMA/RTS/CTS, CSMA/RTS/CTS and CSMA/CA/RTS/CTS to PSMA, CSMA and CSMA/CA. Furthermore, we also evaluate PSMA/RTS/CTS/ACK, CSMA/RTS/CTS/ACK and CSMA/CA/RTS/CTS/ACK against PSMA/ACK, CSMA/ACK and CSMA/CA/ACK. Thus, we are merely affixing RTS/CTS control frames to PSMA, CSMA, CSMA/CA, PSMA/ACK, CSMA/ACK and CSMA/CA/ACK.

With the variable number of hops experiments (Fig. 5a and Fig. 5b), we see that adding RTS/CTS control frames only increase the overhead since there is no channel access contention by competing sources. In fact, the overhead results in a loss of about 80Kbps on average, although the overhead cost becomes minimal as we increase the number of hops.

We now look at the hidden terminal experiments (Fig. 6a and Fig. 6b). Using RTS/CTS with PSMA provides no benefit. However, when combined with CSMA and CSMA/CA, fairness is achieved. The RTS/CTS exchange allows the competing nodes to be aware of each other and thus they are able to better coordinate for equal channel access. With CSMA/RTS/CTS, fairness is achieved without sacrificing the overall throughput of the network. Aggregate throughput is lower with the introduction of RTS/CTS to CSMA/CA. Adding RTS/CTS to PSMA/ACK, CSMA/ACK and CSMA/CA/ACK also aids in fairness. Merging RTS/CTS with PSMA/ACK and CSMA/CA/ACK results in excellent fair sharing between the connection from node 0 to node 1 and node 2 to node 1. Combining RTS/CTS to CSMA/ACK has little effect.

For the exposed terminal experiments (Fig. 7a and Fig. 7b), the addition of the RTS/CTS to PSMA, CSMA and CSMA/CA has a major impact on PSMA but does little for CSMA and CSMA/CA. PSMA/RTS/CTS alleviates most of the capture characteristics of PSMA and at the same time maintains good aggregate throughput. The RTS/CTS acts like a virtual carrier sense for PSMA and thus we expect the performance of PSMA/RTS/CTS to be similar to that of CSMA. This is consistent with the hidden terminal experiments where the virtual carrier sense emulation of the RTS/CTS control frames of PSMA/RTS/CTS behaves like CSMA and thus exhibit similar network behavior as that of CSMA with the hidden terminal environment. When incorporating RTS/CTS with PSMA/ACK, CSMA/ACK and CSMA/CA/ACK, the RTS/CTS control frames again provides better fairness with PSMA/ACK. With just PSMA/ACK, the connection between node 2 to node 3 locks out the connection from node 1 to node 0, achieving a throughput of 1.76Mbps. By introducing RTS/CTS, both sessions are able to more equally share the channel, with the node 2 to node 3 session getting 687Kbps while the node 1 to node 0 achieving 988Kbps. As for CSMA/ACK and CSMA/CA/ACK, adding RTS/CTS results in comparable network performances.

As we move to the ring experiments, we see that total throughput increases with the use of RTS/CTS on top of PSMA, CSMA and CSMA/CA (Fig. 8a and Fig. 8b). Aggregate throughputs of PSMA, CSMA and CSMA/CA in the ring experiment are 3.03Mbps, 2.08Mbps and 2.06Mbps, respectively compared to that of PSMA/RTS/CTS, CSMA/RTS/CTS and CSMA/CA/RTS/CTS whose throughputs are 3.17Mbps, 3.34Mbps and 2.70Mbps, respectively. Using RTS/CTS with PSMA renders the system less fair while the opposite is true with CSMA and
CSMA/CA where the amount of fair sharing is increased. However, when combining RTS/CTS with PSMA/ACK, CSMA/ACK and CSMA/CA/ACK, we observe that the RTS/CTS actually degrades the fair sharing of the system. Thus, too much control frames can be detrimental to the fairing sharing of the wireless channel. Throughput of PSMA/ACK ranges from 446Kbps to 238Kbps while PSMA/RTS/CTS/ACK ranges from 845Kbps to 79Kbps. With CSMA/ACK, minimum and maximum throughput is 444Kbps and 275Kbps, respectively, while CSMA/ACK/RTS/CTS exhibited minimum and maximum throughput of 804Kbps and 37Kbps, respectively. Similarly, CSMA/CA/ACK shows throughput ranging from 555Kbps to 204Kbps while CSMA/CA/ACK/RTS/CTS shows throughput from 566Kbps to 167Kbps.

C. ACK

To determine the effects of ACKs, the following protocols are put in juxtaposition: PSMA/ACK, CSMA/ACK and CSMA/CA/ACK to PSMA, CSMA and CSMA/CA. In addition, we also 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.

We start again with the variable number of hops experiment. From Fig. 9a and 9b, we see as before that the ACKs only adds overhead here since there is no channel contention. The overhead for PSMA/ACK, CSMA/ACK, PSMA/RTS/CTS/ACK and CSMA/CA/RTS/CTS/ACK is approximately 70Kbps compared to that of PSMA, CSMA, PSMA/RTS/CTS and CSMA/RTS/CTS, respectively. However, the overhead for using ACK frames in CSMA/CA/ACK and CSMA/CA/RTS/CTS/ACK is more than 200Kbps. Still, as the number of hops increases, the overhead is reduced substantially.

Examining the results of the hidden terminal experiments (Fig. 10a and Fig. 10b), we learn that ACKs give mixed results regarding the fair sharing of the network. Adding ACKs to PSMA had little effect while adding ACKs to CSMA and CSMA/CA increases fairness, although not totally. When used with PSMA/RTS/CTS, ACKs greatly level the throughput of both connections. However, putting ACKs with CSMA/RTS/CTS actually worsens fair sharing. Using ACKs with CSMA/CA/RTS/CTS added no benefit.

In terms of throughput, when combined with PSMA (1.74Mbps vs. 1.76Mbps), PSMA/RTS/CTS (1.74Mbps vs. 1.78Mbps), CSMA (1.75Mbps vs. 1.99Mbps) and CSMA/RTS/CTS (1740.7bps vs. 1741.2bps), link-level ACKs improve the aggregate throughput while the throughput deteriorates with CSMA/CA (1.72Mbps vs. 1.45Mbps) and CSMA/CA/RTS/CTS (1.61Mbps vs. 1.51Mbps).

In the exposed terminal experiments, link-level ACKs demonstrate no improvements in either fairness or total throughput. As a matter of fact, aggregate throughput decreases slightly (Fig. 11a and Fig. 11b). The lack of improvement when ACKs are introduced is attributed to the fact that in the exposed terminal scenario, link-level ACKs in most cases will not experience any collisions and therefore will not greatly affect the overall performance of the network in the exposed terminal environment. Using RTS/CTS control frames by themselves is sufficient here in the exposed terminal environment.

Moving on to the ring experiments (Fig. 12a and Fig.12b), we see that ACKs with PSMA had little effect. However, with combined with CSMA or CSMA/CA, the overall throughput dramatically increases, from 2.1Mbps in PSMA to 3.2Mbps in CSMA/ACK and from 2.1Mbps in CSMA/CA to 2.7Mbps in CSMA/CA/ACK. The same pattern occurs when we introduce ACKs to PSMA/RTS/CTS, CSMA/ACK/RTS/CTS and CSMA/CA/RTS/CTS, although to a lesser extent. It is uncertain whether or not ACKs have any beneficial outcomes in terms of fairness under the ring experiment.

D. SUMMARY

From the observations in section 4.A, the resolve is that PSMA and CSMA works best overall compared to CSMA/CA in terms of overall network throughput and fairness. By examining section 4.B, we see that RTS/CTS control frames assist to provide fairness to the network and in most circumstances improves the aggregate throughput of the network as well. We also discover that although RTS/CTS control frames when appended to PSMA/ACK, CSMA/ACK and CSMA/CA/ACK in general help reduce capture effects, there are certain circumstances that too much control frames are deleterious to fair sharing. Finally, section 4.C leads us deduce that ACKs in general improve the cumulative throughput of the network environment under study.

The combination that works best together is CSMA/CA/RTS/CTS/ACK when running FTP. CSMA/CA/RTS/CTS/ACK provides a superior mixture of fairness and aggregate network throughput under the four topologies being studied. What distinguishes CSMA/CA/RTS/CTS/ACK from the rest of the protocols we analyze is the fact that even when competing traffics are several hops away from one another, CSMA/CA/RTS/CTS/ACK still manages to provide some sort of channel access coordination among the competing streams (due to lack of space, we are not able to show this particular result).

5. CONCLUSION

From the previous UDP [3] and TCP experiments that we conducted, the general trend is that carrier sensing by itself is superior to packet sensing and carrier sensing with collision avoidance. The RTS/CTS mechanism is mainly responsible for channel access coordination, which in turn
leads to greater fair sharing of the network resource and in some cases improves total throughput. With ACK control frames, more often than not, aggregate throughput is increased due to the persistent retransmission when ACK frames are lost. However, sometimes fairness is sacrificed.

The combination of CSMA/RTS/CTS/ACK is the favorite for UDP traffic with CBR, although CSMA/CA/RTS/CTS/ACK comes in a close second. On the TCP side running FTP, CSMA/CA/RTS/CTS/ACK is the clear winner. Thus, we conclude that CSMA/CA/RTS/CTS/ACK provides the best overall network service under general terms. Not surprising, CSMA/CA/RTS/CTS/ACK is exactly the IEEE 802.11 standard with virtual carrier sense enabled.

ACKNOWLEDGMENT

This research was supported in part by DARPA under contract DAAB07-97-C-D321, by NSF under contract ANI-9814675, by Intel under project "QoS Wireless Networks".

REFERENCE


Fig. 6a. Hidden Terminal, Without ACKs

Fig. 6b. Hidden Terminal, With ACKs

Fig. 7a. Exposed Terminal, Without ACKs

Fig. 7b. Exposed Terminal, With ACKs

Fig. 8a. Ring, Without ACKs

Fig. 8b. Ring, With ACKs

Fig. 9a. Variable Number of Hops, Without RTS/CTS

Fig. 9b. Variable Number of Hops, With RTS/CTS

Fig. 10a. Hidden Terminal, Without RTS/CTS

Fig. 10b. Hidden Terminal, With RTS/CTS

Fig. 11a. Exposed Terminal, Without RTS/CTS

Fig. 11b. Exposed Terminal, With RTS/CTS

Fig. 12a. Ring, Without RTS/CTS

Fig. 12b. Ring, With RTS/CTS