TCP Performance in Wireless Multi-hop Networks

Mario Gerla, Ken Tang, Rajive Bagrodia
{gerla, ktang, rajive}@cs.ucla.edu
Wireless Adaptive Mobility Laboratory
Computer Science Department
University of California, Los Angeles
Los Angeles, CA 90095
http://www.cs.ucla.edu/NRL/wireless

Abstract

In this study we investigate the interaction between TCP and MAC layer in a wireless multi-hop network. Using simulation, we provide new insight into two critical problems of TCP over wireless multi-hop. The first is the conflict between TCP data packets and TCP ACKs, which causes performance to degrade for window sizes greater than 1 packet. The second is the interaction between TCP and MAC layer backoff timers which causes severe unfairness and capture conditions. We show that these effects are particularly pronounced in two families of MAC protocols that have been extensively used in ad-hoc network simulation and implementations, namely CSMA and FAMA (a descendent of MACA). We then demonstrate that these problems are in part overcome by using MACAW, a MAC layer protocol which extends MACA by adding link level ACKs and a less aggressive backoff policy. We argue that link level protection, backoff policy and selective queue scheduling are critical elements for efficient and fair operation of ad-hoc networks under TCP. These conclusions are supported by extensive simulation and measurement experiments.

1. Introduction

The rapid advancement in portable computing platforms and wireless communication technology has led to significant interest in the design and development of protocols for instantly deployable, wireless networks often referred to as "ad-hoc networks". Ad-hoc networks are required in situations where a fixed communication infrastructure, wired or wireless, does not exist or has been destroyed. The applications span several different sectors of society. In the civilian environment, they can be used to interconnect workgroups moving in an urban or rural area or a campus and engaged in collaborative operation such as distributed scientific experiments and search and rescue. In the law enforcement sector, applications such as crowd control and border patrol come to mind. In the military arena, the modern communications in a battlefield theater require a very sophisticated instant infrastructure with far more complex requirements and constraints than the civilian applications [8].

Two key requirements of the ad-hoc network environment are reliable data transfer and congestion control. These features are generally supported by TCP. An important question is how TCP (which has been designed and fine-tuned for wired networks) interacts with the wireless protocols, in particular the MAC layer. Both MAC and TCP layers strive to provide efficient transport in a shared environment, with some degree of efficiency and with protection from errors and interference. The MAC layer however has only a myopic view of the network, which is a critical limitation in multi-hop networks. In contrast, TCP provides a true end-to-end control on errors and congestion.

An important issue, which we address in this paper, is wireless link error and loss protection using logical link ACKs. There is a vast literature showing the benefits of link level ACKs in conjunction with TCP error and window control [6, 14]. However, most of the studies were based on cellular type, single hop systems. In this study, we address multi-hop networks and thus must deal with much more complex collision patterns (e.g., hidden terminals) than in cellular type systems.

In this paper, we study the TCP/MAC layer interaction via simulation. The simulation platform used is GloMoSim [13]. GloMoSim is a parallel simulation environment implemented in PARSEC, PARallel Simulation Environment for Complex Systems [1]. It includes several wireless protocols in its library (radio propagation, mobility, MAC, network, transport and applications). 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.

The rest of the paper is organized as follows: Section 2 reports the configuration and parameters we used for our simulation. TCP over the MAC layer experiments are examined in section 3. Section 4 summarizes our grid topology simulation results. Validation of our simulation results with actual TCP testbed measurements using WaveLAN radios is provided in the section 5. Finally, section 6 concludes the paper.

2. Experimental Configuration and Parameters

We consider three different types of topologies: a string topology with 8 nodes (0 through 7) as shown in Figure 1; a ring topology with 8 nodes as shown in Figure 2; and a grid topology with up to 100 nodes as shown in Figure 3. Radio channels are bi-directional; the arrows indicate the direction of data packet transmissions. ACKs travel in the opposite direction. The distance between two neighbor nodes is equal to the radio transmission range. TCP or UDP connections are established between different node pairs. These connections carry large file transfers (i.e. infinite backlog). In some cases (grid topology), they carry interactive traffic. Nodes are static (no mobility). Free space channel model is used in the simulation model. Perfect channel is assumed (no external noise). Channel data rate is 2Mbps.

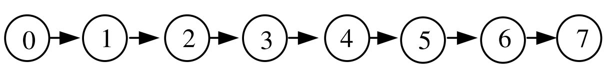

Figure 1. String Topology.

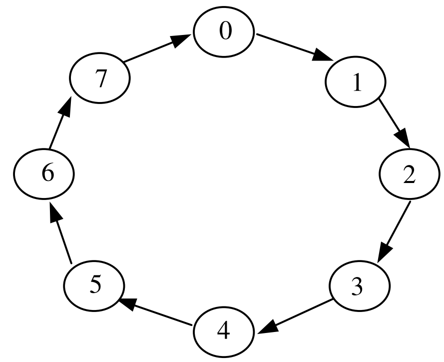

Figure 2. Ring Topology.

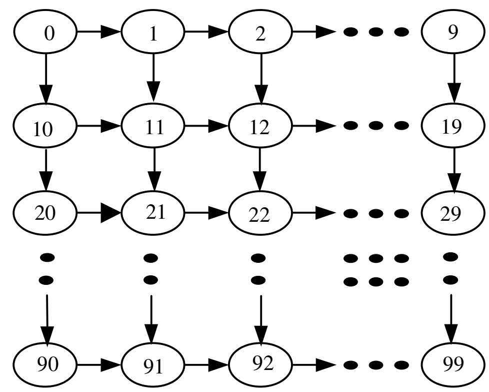

Figure 3. Grid Topology.

Three MAC protocols are considered: CSMA, FAMA and MACAW. These protocols were chosen because they are representative of a broad class of MAC schemes used in wireless LANs. 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" problem. Nodes 2 and 4 (in Figure 1) cannot hear each other but are both within reach of node 3. They may transmit simultaneous (in spite of carrier sensing) and thus cause a collision at node 3. CSMA was used first in the Packet Radio network in the mid 1970's [9]. FAMA (Floor Acquisition Multiple Access) is an experimental MAC protocol specifically developed for the Glomo DARPA program. It is a descendent of MACA (Multiple Access Collision Avoidance) [10]. Like MACA, it uses the RTS (Request To Send) and CTS (Clear To Send) exchange to prepare the floor for data transmission (thus avoiding "hidden terminal" collision in most cases) [7]. However, unlike

MACA, it also uses carrier sensing. MACAW [5] extends MACA by introducing additional control packets (DS, ACK, and RRTS). A typical frame exchange between node 1 and node 2 starts with an RTS from node 1. Node 2 then responds with a CTS. Node 1 in turn sends a DS immediately followed by the data frame. Lastly, node 2 acknowledges the data frame with an ACK. If node 2 is in the QUIET state and receives an RTS from node 1, node 2 sends an RRTS to node 1 as soon as node 2 becomes idle. After receiving the RRTS, node 1 starts the regular frame exchange process. In addition, MACAW abandons the exponential backoff used in most random access schemes and instead incorporates a slightly different backoff scheme called MILD backoff. Also, a backoff-copying scheme is adopted to ensure that the contending nodes use the same backoff value. Furthermore, a multiple stream model is included to improve fairness.

Each node has a 25-packet MAC layer buffer pool. Incoming packets that find the buffer full are dropped. Scheduling of packet transmissions is FIFO. The routing protocol is Distance Vector (Bellman-Ford). In order to avoid interference between routing update packets and TCP packets, the Distance Vector algorithm is run only at the start of each experiment to initialize the routing tables. This is made possible in our case by the fact that nodes are stationary and topology does not change during the experiments.

The TCP simulation model is an accurate replica of the TCP code running in the Internet hosts today. The TCP simulation code was generated from FreeBSD 2.2.4 code. In particular, window size grows progressively until it reaches the advertised window or until packet loss is detected. In the latter cases, window size is halved (fast retransmission and fast recovery) or abruptly reduced to 1 (slow start). In some of our experiments, we will "force" the maximum TCP window to take at a certain value by setting the advertised window to such value (e.g., 1460B). TCP packet length is assumed fixed at 1460B. In some experiments, packet loss due to channel interference is so high that some TCP connections are timed out and closed. To allow the simulator to run to completion, we artificially increased the maximum number of retransmissions before shutoff, thus avoiding premature TCP connection closure.

In this paper, we have chosen topologies and configurations that are quite simplistic (for example, ring, string and grid. Also, the distance between the nodes is assumed to be equal to the power range of each node. Furthermore, only FTP and interactive traffic are considered. These simplistic assumptions allow us to better isolate, analyze and understand the specific, pathologic patterns of behaviors such as capture and fairness. Once these behaviors are understood, we plan to extend our experiments to incorporate more realistic situations that are representative of the real world, such as more pragmatic topologies, mobile nodes, varying node distances and traffic traces such as World Wide Web or tcplib. We will also include the IEEE 802.11 MAC layer standard in our comparison.

3. TCP over MAC Layer

Previous MAC layer simulation experiments at UCLA [2] have uncovered two major weaknesses of the MAC layer: (a) CSMA and, to a lesser extent, FAMA suffer from the hidden terminal losses; (b) one or more stations tend to "capture" the channel in heavy load situation. The capture phenomenon is also reported in [5] and [15]. Thus, loss recovery must be provided by a layer above MAC, either the link layer or the transport layer. In [2], we considered loss recovery only at the TCP transport layer protocol, as it is the most popular solution in wireless LANs and it is the easiest to implement. We also investigated the effectiveness of TCP in removing MAC layer capture. The first set of results we obtained from these experiments was rather negative. TCP alone does not improve throughput performance in most cases. In some cases it makes it worse. For example, in multi-hop wireless paths TCP on top of CSMA and FAMA alike dramatically degrades performance when the congestion window is greater than 1 packet. TCP mitigates capture behavior in some cases, but makes it worse in others [2]. In conclusion, we realized that additional techniques (beyond mere TCP) are necessary to obtain good performance in a multi-hop wireless environment.

As previously indicated, one possible technique is the use of link level ACKs. However, the introduction of an additional layer implies, in general, considerable processing and control traffic overhead (besides the cost of the implementation). Therefore, we opted initially for a simpler solution. That is, we embedded the ACK in the MAC layer protocol. This technique, initially introduced in Ethernet as an option, does in fact exist in MACAW as well as in the IEEE 802.3 standard. Thus, we proceeded to carry out a similar set of experiments as those described in [2], this time with MACAW. We also increased the size of the networks under study and expanded the gamut of traffic types and patterns, in order to verify the validity of the results in more general environments.

In this paper, we report the results of our MACAW experiments. For completeness, we also include a summary of the previous results, so that the comparison between the various protocols can be easily made.

We start by examining the interplay of TCP and MAC layers when there is only one TCP connection present in the network. More precisely, we consider a single TCP connection that covers a variable number of hops, from 1 to 7. In the first set of experiments, TCP window (W) is kept fixed to the value 1460B (i.e., W = 1 packet). The results for CSMA, FAMA and MACAW throughputs as a function of number of hops H are reported in Table 1. One can verify that throughput values match exactly the analytic predictions for a send-and-wait protocol. The throughput is inversely proportional to the hop distance. CSMA throughput is slightly higher than FAMA [12] because of RTS/CTS overhead in the latter. MACAW exhibits the lowest throughput among the three protocols due mostly to the additional control frames.

Number of Hops	CSMA	FAMA	MACAW
1	1838.4	1476.5	971.7
2	921.3	718.7	485.8
3	614.8	475.4	323.9
4	461.4	355.3	242.9
5	369.2	287.5	194.3
6	307.7	239.1	161.9
7	263.4	204.7	138.8

Table 1. Throughput (Kbps), Single TCP Connection, Variable Number of Hops, W = 1460B.

Next, we allow the TCP window to grow up to W = 32KB. Namely, the TCP protocol dynamically adjusts the congestion window as required. As the window is increased, multiple packets and multiple ACKs travel on the path in the opposite direction, creating interference and collisions. We would expect that in balance the window increase improves performance since for 7 hops, for example, analysis shows that the optimal throughput (assuming optimal scheduling of packet and ACK transmissions along the path) is achieved for W = 3 x 1460B. The simulation results in Table 2 indicate that this is not true for CSMA and FAMA. CSMA throughput collapses when H > 2. Hidden terminal losses, which become very substantial for longer paths, cause the loss of TCP ACKs with consequent timeouts and major throughput degradation. FAMA does slightly better than CSMA, yet it is not totally immune from hidden terminal effects. This clearly shows for longer paths (H > 3), where the throughput is much lower than that achieved with W = 1460B. MACAW, on the other hand, fares quite well with larger TCP windows. In fact, the throughput is consistently higher than for W = 1 packet. Note that the throughput tends to become constant as hop distances grow larger - a typical sign of effective pipelining. Moreover, the asymptotic value of 300Kbps is not too far off from the theoretical maximum achievable on a multi-hop wireless path, which is 25% of channel capacity (in our case 0.25×2000Kbps = 500Kbps). The good performance of MACAW is attributed to the fact that MACAW acknowledges every frame sent and performs local retransmissions if an acknowledgement for a frame is not received. This precludes TCP ACK loss and consequent TCP timeouts. An interesting anomaly is noticed for the single hop connection where the large window does slightly worse than W = 1460B. This is explained by the fact that no pipelining advantages can be obtained in this case. Yet, the TCP ACKs in the opposite direction compete for the same channel and steal some of the bandwidth (note that with W = 1 packet, there is no contention since packet and ACKs take turns in using the channel).

Number of Hops	CSMA	FAMA	MACAW
1	1791.2	1458.7	888.8
2	439.5	716.2	612.3
3	0.5	389.4	411.2
4	0.5	71.4	364.8
5	0.5	13.5	335.0
6	0.5	72.8	324.9
7	0.5	66.9	311.7

Table 2. Throughput (Kbps), Single TCP Connection, Variable Number of Hops, W = 32KB.

From the above results we conclude that if the wireless link is not protected by a link level ACK (as is the case of CSMA and FAMA), it is counterproductive to use W larger than single packet size even on connections covering multiple hops. The larger window, on the other hand, is very effective on multiple hops when the link level ACK is present, as in MACAW.

In the next experiment, we study the interaction of several TCP connections, each covering just one hop. More specifically, we consider the ring topology in Figure 2. The 8 nodes are engaged in single hop file transfer connections (0-1, 1-2, etc). The results for CSMA, FAMA and MACAW are reported in Table 3. We only consider W = 1460B since W > 1460B gives no improvements in a single hop environment. The performance measures of interest in this experiment are throughput efficiency and fairness.

TCP Connection	CSMA	FAMA	MACAW
0-1	275.0	34.9	171.9
1-2	298.7	34.7	174.1
2-3	180.5	28.4	172.4
3-4	400.8	0.1	173.3
4-5	426.5	1472.6	173.7
5-6	253.6	1.2	173.4
6-7	307.1	0.2	174.3
7-0	362.8	1506.9	172.2
Total Throughput	2505.0	3079.0	1385.3

Table 3. Throughput (Kbps) in Ring Topology, W = 1460B.

We start with CSMA and note that the behavior displays some capture. The throughput of connection 2-3 is less than half that of 4-5. Aggregate throughput is 2.5Mbps, which 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). With FAMA, TCP yields aggregate throughput of 3Mbps! This compares very favorably with the theoretical maximum of 4Mbps. However, this solution is unacceptably unfair. Two links capture the entire throughput. The remaining links have small to negligible throughput. The capture behavior is explained by the interplay of CSMA and FAMA 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. In contrast with the previous MAC protocols, MACAW is extremely fair, with each connection getting a throughput of approximately 170Kbps. The fair behavior is due to the MILD (instead of binary) backoff, and to the recovery from losses. However, the aggregate throughput is only 1.4Mbps, the lowest among the three MAC layer protocols. We note here a phenomenon quite common in multi-access networks, and more generally in resource sharing systems. In order to obtain maximum total throughput, it is often better to select and serve only a few users, so that the control and probing overhead required for fair sharing is eliminated. Capture is an example of this principle. Thus, it is no surprise that the higher the capture, the better the total throughput.

Next, we wish to examine the behavior of a system which combines both single and multi-hop sessions. TCP multi-hop sessions have the tendency to suffer unfair degradation even in wired networks. We expect this unfair treatment to be further amplified by the wireless channel. To this end, we consider the case of a TCP connection spanning several hops sharing the path with several single hop connections. The experimental configuration (see Figure 4) consists of a string with 8 nodes, seven single hop connections (0-1, 1-2, etc) and one multi-hop connection (0-7). The results are reported in Table 4. We find zero or negligible throughput for the (0-7) connection for both CSMA and FAMA. This is not unexpected since, without link level ACKs, the probability of a packet making it through 7 hops in the face of contention from single hop connections is indeed slim! As for the single hop sessions, an interesting observation is that in this case CSMA exhibits much stronger capture (and thus higher overall throughput) than FAMA, exactly the opposite of what was happening in the ring experiment in Table 3. Clearly, capture prevents reaching a steady state within the duration of the simulation experiment and thus leads to unpredictable results. With MACAW, things are slightly better for the multi-hop TCP connection. Throughput climbs to 16Kbps with the help of ACKs and local retransmissions. We notice also that some capture is now present even in MACAW. Nodes at either end of the string have higher throughput than the rest because they only compete with one neighbor. What is more surprising, however, is the fact that the middle link (3-4) gets a throughput almost 3 times that of other internal nodes. This show that link level ACKs and MACAW's backoff schemes are by no means the complete solution for fairness, and therefore other schemes must be also be considered (e.g., queue scheduling).

TCP Connection	CSMA	FAMA	MACAW
0-1	358.7	334.2	442.9
1-2	20.6	437.5	116.5
2-3	811.3	336.9	132.1
3-4	960.7	147.0	291.9
4-5	21.8	834.0	103.5
5-6	0.0	248.9	168.1
6-7	1630.1	387.7	567.8
7-0	0.0	0.1	16.3
Total Throughput	3803.3	2726.3	1839.1

Table 4. Throughput (Kbps) in String Topology with 0-7 Data Stream, W = 1460B.

It is interesting to compare the behavior of the single hop connections in the string versus the ring topology. Unfairness and capture are typically more pronounced in the string than in the ring, mainly because of border effects. Aggregate throughput is better in the string than in the ring. This is expected since the maximum theoretical throughput on the string (based on idealized scheduling of transmissions) is 6Mbps, versus 4Mbps on the ring topology.

In summary, the following lessons were learned from the TCP experiments in heavy file transfer load. First of all, a large window has a negative effect in absence of link loss protection. A window of one packet size provides by far the best results for CSMA and FAMA. With link protection, like in MACAW, throughput is substantially improved on multi-hop paths. Secondly, we found that TCP does not alleviate capture for CSMA and FAMA. We recall that CSMA and FAMA are prone to capture even under UDP, without TCP [2]. Our results show that TCP tends to amplify the problem. Apparently, the backoffs in CSMA/FAMA and TCP reinforce each other, emphasizing capture and unfairness. Reasonably fair behavior is achieved with MACAW (especially in the ring topology) due to link level ACKs and careful timeout design.

4. Grid Topology

The previous experiments have been based on linear topologies (string or ring) with just one type of traffic (file transfers). Linear topologies are very helpful in interpreting the simulation results, but are not very common in ad-hoc network environments. In this section, we consider a more realistic environment with a much richer traffic mix, topology connectivity and number of nodes, and verify that the results observed with linear topologies apply also to this more general case. To this end, we have selected a 100-node grid topology where each node has four neighbors except for the nodes positioned at the edge of the grid (see Figure 3). Two types of traffic are injected in this network: FTP traffic along all vertical paths (i.e., from 0 to 90, from 1 to 91, etc), and interactive traffic along all horizontal hops (i.e., from 0 to 1, from 1 to 2, from 2 to 3, ... from 10 to 11, etc). The interactive traffic is modeled by a constant offered rate of fixed size packets. The rate is uniform over the network and varies from experiment to experiment. File transfer sources have constant, heavy supply of packets to transmit. All sessions, interactive and FTP, are controlled by TCP.

The performance measure of interest is again throughput. The key variables are offered interactive load and network size. We conduct experiments with the following loads (per interactive TCP connections): 1.5Kbps, 23.3Kbps and 233.6Kbps.

We begin with the analysis of the CSMA experiment with offered load of 1.5Kbps (Figure 4). The interactive sources achieve their full throughput; they manage to discharge all packets that arrive. The FTP sources fare less well. They achieve throughputs ranging from 12Kbps to 64Kbps, with average of 31.9Kbps (see Figure 4). The theoretical maximum is 200Kbps (9 hops, W = 1 packet). The gap between theoretical maximum and measured values is due to interference from neighbor FTP connections as well as from interactive traffic on the cross-links. Particularly damaging to multi-hop FTP connections is the high loss rate on the links and the lack of link loss recovery. The FTP throughput at the borders of the grid (i.e., 0 to 90 and 9 to 99) is much higher than in the interior since the interference is lower there. Unfairness due to capture is already quite evident, even at this low level of interactive traffic.

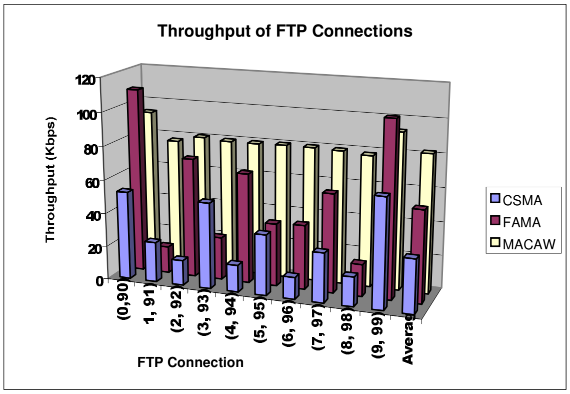

Figure 4. 10 X 10 Grid Topology, Offered Interactive Load of 1.5Kbps.

Next, we shift our attention to FAMA for the same interactive load (see Figure 4). Interactive connection performance is about the same as with CSMA. FTP average FAMA throughput is higher than CSMA, 54.5Kbps as compared with 31.9Kbps in CSMA. FAMA is better protected against hidden terminal losses. This reduces FAMA losses and improves its throughput. Unfairness and capture, however, are present also in FAMA (maximum of 109Kbps, minimum of 15Kbps). As already noted before, unfairness is due to the progressively increasing random timeouts.

With MACAW, interactive throughput is same as CSMA and FAMA. However, the average FTP throughput is higher than both CSMA and FAMA, with an average throughput of 81.9Kbps. The higher throughput is the direct result of MACAW's local retransmissions and MILD backoff algorithm. MACAW is also very fair due to its backoff copying scheme and separate scheduling of data streams directed to different neighbors (this feature was not exploited in previous experiments since all traffic was forwarded through the same neighbor). The only (unavoidable) unfairness is found at the boundary sessions.

As the interactive offered load is increased to 23.3Kbps (see Figure 5), we start noticing a slight loss in the interactive throughput for CSMA and FAMA. Namely, the interactive sources cannot keep up with the offered load. In the CSMA experiment the interactive throughput is 20.7Kbps. In FAMA, it is 21.4Kbps. With MACAW, it is 23.4Kbps. Thus, the MACAW protocol can sustain the full interactive load. As for the FTP connections, the increase in interfering load causes strong degradation and strong capture, especially for CSMA. Maximum and minimum values in CSMA are 6.1Kbps and 0.04Kbps. Some FTP connections are practically locked out! FAMA fares a bit better, with maximum and minimum of 22.9Kbps and 1.9Kbps, respectively. MACAW is by far the best performer, with maximum of 79.1Kbps and minimum of 53.5Kbps.

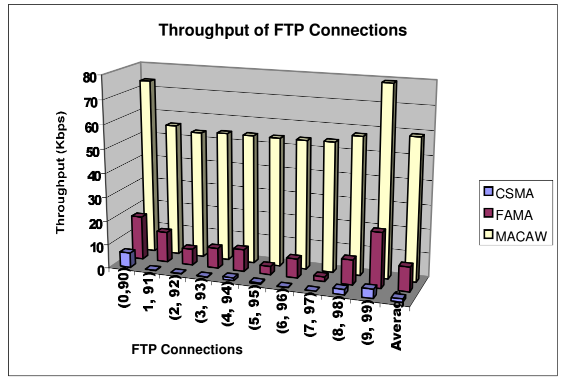

Figure 5. 10 X 10 Grid Topology, Offered Interactive Load of 23.3Kbps.

For interactive load of 233.6Kbps (see Figure 6), the interactive sources show strong signs of capture/unfairness. In CSMA, we measure a maximum of 121Kbps and minimum of 11.4Kbps for interactive throughput. With FAMA, maximum and minimum are 180.5Kbps and 36Kbps. MACAW has a maximum and minimum of 186.1Kbps and 19.7Kbps. Average interactive connection throughput is 53Kbps in CSMA, 73.8Kbps in FAMA and 87.6Kbps in MACAW. As for the FTP throughput, it virtually collapses at this load for CSMA and FAMA. Both show same average throughput of 0.087Kbps. MACAW shows a surprisingly uniform throughput for the intermediate FTP connections (5Kbps). The border connections fare much better, of course. Average FTP throughput is 8.1Kbps.

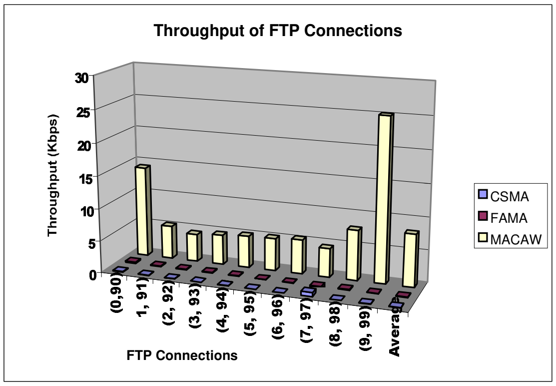

Figure 6. 10 X 10 Grid Topology, Offered Interactive Load of 233.6Kbps.

In summary, the grid experiments confirm the behavior observed in linear topologies. Namely, poor multi-hop connection performance under TCP and heavy capture for both CSMA and FAMA. FAMA performs consistently better than CSMA. It is clear, however, that neither CSMA nor FAMA are adequate for ad-hoc network operation without the assistance of link level loss protection and of proper fairness measures. MACAW is by far superior to both CSMA and FAMA in terms of throughput and fairness. The grid experiment, because of cross traffic, exercises and exploits an additional fairness feature of MACAW, namely the ability to separately schedule transmissions to different neighbors, and thus more fairly allocate the channel among them [5]. In spite of this feature, MACAW still exhibits capture and unfairness among interactive sources for high interactive traffic rates.

5. TCP Measurements

In order to validate the simulation results with testbed measurements, we set up a 4-node ring configuration using WaveLAN radios and Linux laptops. To enforce a ring configuration, four laptops were placed at each corner of Boelter Hall at UCLA: the bulk of Boelter Hall precluded cross links. Our WaveLANs (old WaveLANs that are not IEEE 802.11 compliant) use CSMA MAC layer. The Linux laptops run TCP on top of the WaveLAN interface. TCP segment size was set to 1460B. TCP window size is dynamically adjusted up to the advertised window, which is 32KB. Table 5 and Table 6 report measurements and simulation results, respectively. Measurements were repeated three times, and so were the simulation experiments. Repetition is important since we know that the system exhibits capture, thus the "capturing" node or nodes may change from run to run.

TCP Connections	Trial 1	Trial 2	Trial 3
0-1	15.22	578.51	574.34
1-2	9.53	24.96	7.22
2-3	990.09	130.40	102.75
3-0	63.33	42.91	1.03
Total Throughput	1078.16	776.78	685.34

Table 5. Throughput (Kbps) in 4-Node Ring Topology, WaveLAN Measurements.

TCP Connections	Trial 1	Trial 2	Trial 3
0-1	292.7	192.2	258.2
1-2	111.3	263.6	11.0
2-3	26.8	64.7	8.0
3-0	105.4	29.0	365.3
Total Throughput	536.3	549.5	642.5

Table 6. Throughput (Kbps) in 4-Node Ring Topology, GloMoSim Simulation, W = 32KB.

The WaveLAN measurement runs in Table 5 exhibit the same unfairness/capture behavior observed in the simulation experiments in Table 6. In the WaveLAN experiments, some of the TCP connections even got disconnected. The simulation aggregate throughput is lower than the measured throughput. This is due in part to the fact that the WaveLAN CSMA implementation uses different timeouts than the simulator. Furthermore, the TCP simulation model does not include the disconnect feature, which is triggered when the number of retransmissions reaches a threshold value. The disconnect feature eliminates some of the competitors, thus reducing interference and increasing total throughput. The analysis of simulation and measurement traces reveal that connections take turn in transmitting packets. For instance, in experiment 1 in Table 5, only connection 2-3 gets going. In experiment 3, connection 2-3 starts first, and then connection 0-1 takes over. Between the "capture" states, there are intervals during which connections compete for the channel and little traffic gets through. This behavior is confirmed by analysis. One can easily verify that a 4-node ring can only support a single packet transmission between node pairs at a time. Two simultaneous transmissions will collide.

6. Conclusion

The focus of the paper has been the MAC/TCP layers interaction in a multi-hop radio network. We have considered three representative MAC layers: CSMA, FAMA and MACAW. The main findings of the study are:

(a) Both CSMA and FAMA exhibit capture under TCP. Namely, connections take turns in "capturing" the channel. The capture may last several seconds, even minutes. This is clearly unacceptable in real time environments such as the battlefield. The problem was traced to the lack of link loss protection, as well as to the aggressive binary backoff policy. MACAW, with its ACK feature, MILD backoff and selective scheduling provides much better fairness, although at the expense of lower throughput. The lower throughput is particularly evident in isolated, single hop TCP connections.

(b) Multi-hop TCP connections in wireless networks are at a clear disadvantage with respect to single hop connections, much more so than in wired networks. In case of channel contention, MAC protocols without link loss protection (i.e., CSMA and FAMA) yield zero throughput on multi-hop connections. Moreover, the best performance is achieved with W = 1 packet (to avoid conflict between multiple packets and outstanding TCP ACKs). Again, MACAW performs much better also on multi-hop connections because it eliminates packet drop (due to hidden terminals) and thus allows the TCP connection to operate with a window greater than 1 packet.

The main message derived from this study is that CSMA and FAMA protocols in their current version are not adequate to support ad hoc-networks involving TCP applications with potentially heavy traffic (e.g., FTP). As a minimum, link loss protection must be provided. Furthermore, the exponential (or binary) backoff strategy is suspect. Alternative schemes, such as the MILD scheme in MACAW, should be explored to avoid capture. Likewise, some form of class-based queueing and scheduling (e.g., separate queue per neighbor like in MACAW) should be implemented to improve fairness.

The results indicate that more research is necessary to make TCP and MAC layers work consistently well together in a multi-hop environment. More precisely:

(a) More work on link loss protection. Is it better to embed the ACKs in the MAC layer (like in MACAW), or should one consider a separate Data Link layer (with richer features)?

(b) More work on MAC backoff strategies and on their interplay with upper layer backoff schemes (TCP and others).

(c) More work on queue scheduling within the wireless node.

(d) The interplay between backoff policy, link protection scheme, per queue transmission scheduling and congestion control within the MAC layer must be further investigated. For example, which of the above features is really responsible for improving performance in any of the situations considered in this study? Or, do we need all of them together?

(e) Are additional features required at the transport level (such as the SNOOP proxy [4]) in order to improve performance on top and above what is achieved with MAC and link layer fixes? In the limit, should one consider modifications to the TCP protocol remedies?

References

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

[2] R. Bagrodia and M. Gerla, “A Modular and Scalable Simulation Tool for Large Wireless Networks”, International Conference on Modeling Techniques and Tools for Computer Performance Evaluation, 1998.

[3] Bikram Bakshi, Krishna, P., Pradhan, D.K., and Vaidya, N.H., “Performance of TCP over Wireless Networks”, 17th Intl. Conf. on Distributed Computing Systems (ICDCS), Baltimore, May, 1997.

[4] Hari Balakrishnan, Srinivasan Seshan, Randy H. Katz, “Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks”, ACM Wireless Networks, 1(4), December 1995.

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

[6] David A. Eckhardt and Peter Steenkiste, “Improving Wireless LAN Performance via Adaptive Local Error Control”, INCP, 1998.

[7] 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.

[8] M. Gerla and J.T.-C. Tsai, “Multicluster, mobile, multimedia radio network”, ACM/Baltzer Journal of Wireless Networks. vol. 1, (no. 3), 1995, p. 255-265.

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

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

[11] Robert Morris, “TCP Behavior with Many Flows”, International Conference on Network Protocols, October 1997.

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

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

[14] M. Zorzi and R.R. Rao, “Error control in multi-layered stacks”, Proceedings of Globecom ’97, November 1997.

[15] M. A. Marsan and Mario Gerla, “Fairness in Local Computer Networks”, ICC Conference Proceedings, June 1982.