IEEE 802.11
Wireless LAN Medium Access Control and Physical Layer Specifications

RF MAC Simulation Highlights

10 January 1994

Carlos Puig
Apple Computer, Inc.
One Infinite Loop, MS 301-4J
Cupertino, CA 95014

Issues Addressed

29.1 How does 802.11 address simulation?

1. Introduction

Apple's RF MAC Simulator (RFMACSIM) was developed to help wireless LAN designers evaluate the strengths and weaknesses of four MAC protocols. This paper presents some key insights gained from initial work with the simulator at Apple during July 1993. A detailed description of the simulator's operation and use is provided by a companion report [1]. The introductory section from that report is reproduced in the Appendix.

The following abbreviations refer to the four main protocols[1] simulated by RFMACSIM:

C	= DATA
C+A	= DATA + ACK
R/C+C	= RTS/CTS + DATA
R/C+C+A	= RTS/CTS + DATA + ACK (4-Way)

The initial simulations focused on the relative performance of these four protocols under various assumptions about network geometry, receiver characteristics, and traffic patterns.

[1]RFMACSIM can also simulate ALOHA, but no ALOHA results are included in this paper.

Sample simulation results are included to illustrate the qualitative relationships discussed in each section. These results are drawn from a variety of simulations, based on widely differing parameter sets. Therefore, comparisons among the results in different sections should not be used to draw general, quantitative conclusions about the relative performance of the featured MAC protocols.

2. Network Geometry

The station arrangement, or network geometry, determines the extent to which the "hidden terminal" problem is present. In our simulations, MAC protocol performance was simulated for the following three network geometries:

- Normal case: 8 stations in a 4 x 2 rectangle, with each station in a 10 m x 10 m cell.
- Hidden nodes case: 8 stations in a 4 x 2 rectangle, with each station in a 100 m x 50 m cell.
- Heavy interference case: 16 stations in an 8 x 2 rectangle, with each station in a 50 m x 50 m cell.

In the normal case, all stations can easily receive and sense each other's transmissions. The hidden node case includes several pairs of stations that are too distant to sense each other's transmissions. In the heavy interference case, the hidden node problem is very severe.

In the normal case, network throughput is highest for the simplest protocol, C, and decreases as more handshaking overhead is added (C+A, R/C+C, and R/C+C+A in last place). In the hidden nodes and heavy interference cases, the relative performance of the MAC protocols depends on factors such as the data packet length and receiver RF parameters. These tradeoffs are discussed in the following sections.

3. Acceptance vs. Completion Rates

Although network throughput has traditionally been used as the principal index of network performance, the simulations suggest that throughput may not be sufficient to establish the relative merits of two protocols. Important insights are gained by considering throughput in terms of acceptance and completion rates.

Define a data *block* as a byte array offered to a station's MAC for transmission. I use "block" rather than "packet" to avoid confusion with the transmitted data packet. When a block is offered to the MAC for transmission, the block is immediately *accepted* or rejected. Blocks are rejected when the source station's queue is full. An accepted block that successfully passes through all of the MAC protocol steps has *completed* the protocol. A block may not complete the protocol for a variety of reasons: too many access attempts, a collision for which no retry is allowed, etc. Based on these concepts, we define the *acceptance rate* as the ratio of accepted blocks to completed blocks, and the *completion rate* as the ratio of completed blocks to accepted blocks. The *offered load* is simply the rate at which blocks are offered to the MAC layer, and *throughput* is the rate at which blocks complete the protocol. These definitions allow us to decompose throughput into the product:

Throughput = (offered load) x (acceptance rate) x (completion rate)

Thus the same relative throughput can be realized by (a) a high acceptance rate with a low completion rate, or (b) by a low acceptance rate with a high completion rate.

In the hidden nodes and heavy interference cases, ACK-based protocols (C+A and R/C+C+A) tend to have high completion rates and low acceptance rates. On the other hand, the non-ACK protocols (C and R/C+C) tend to have low completion rates and high acceptance rates. For these network geometries, the simple C protocol typically has much higher throughput at high offered loads than any of the others. However, the high throughput is obtained with a very low completion rate (typically under 50%). The C protocol tries only once to send data over difficult, interference-ridden paths, and therefore, ends up transferring data mostly between nearest neighbors, who are relatively immune from hidden node interference. On the other hand, ACK-based protocols tenaciously continue to retry a difficult path, while refusing to accept additional packets for transmission.

Figures 1 through 3 illustrate the relationship among the acceptance rate, completion rate, and network throughput for the heavy interference network geometry. In this simulation series, completion rates fall far short of 100%, because a relatively small maximum retry limit was used.

To the extent that a "delivery guarantee" at the MAC layer is important to network service quality, MAC protocols should be compared in terms of throughput only when they achieve similar completion rates. Because of the importance of an ACK for achieving high completion rates, much of our later simulation work focused on the two ACK-based protocols.

Figure 1. Acceptance Rate For Heavy Interference Case.

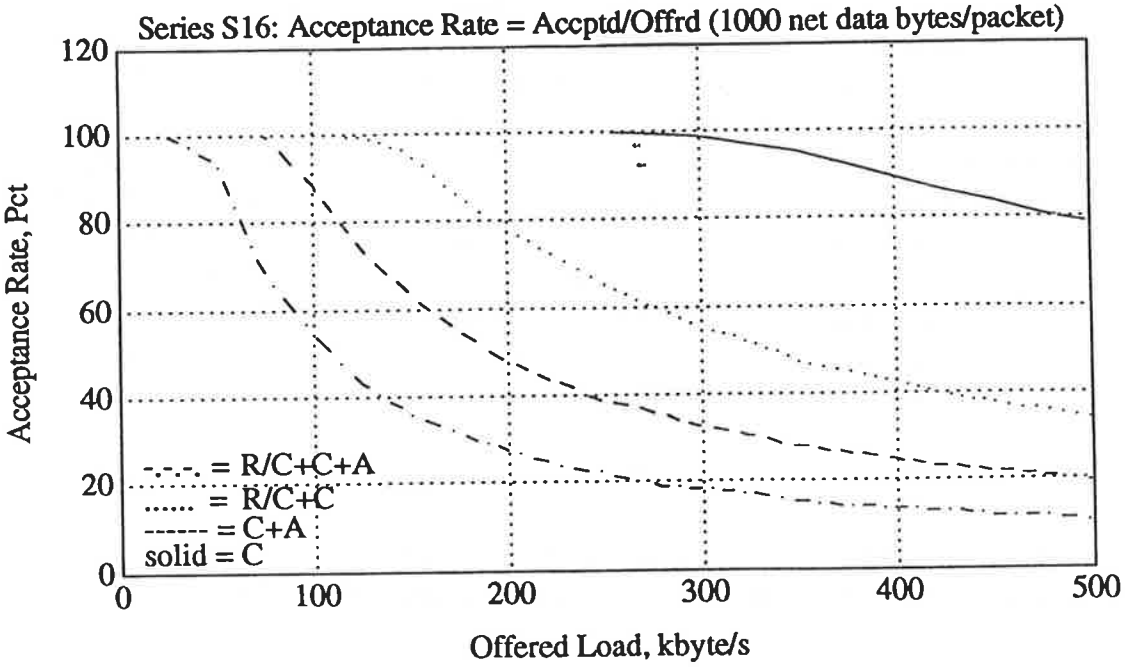


Figure 2. Completion Rate For Heavy Interference Case.

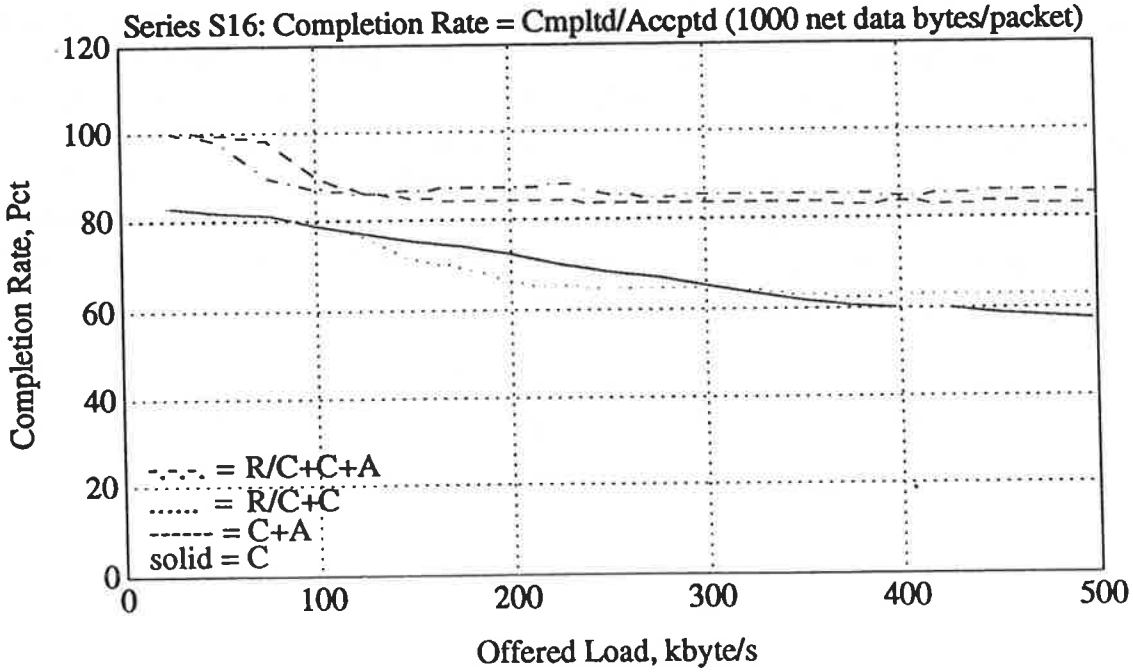
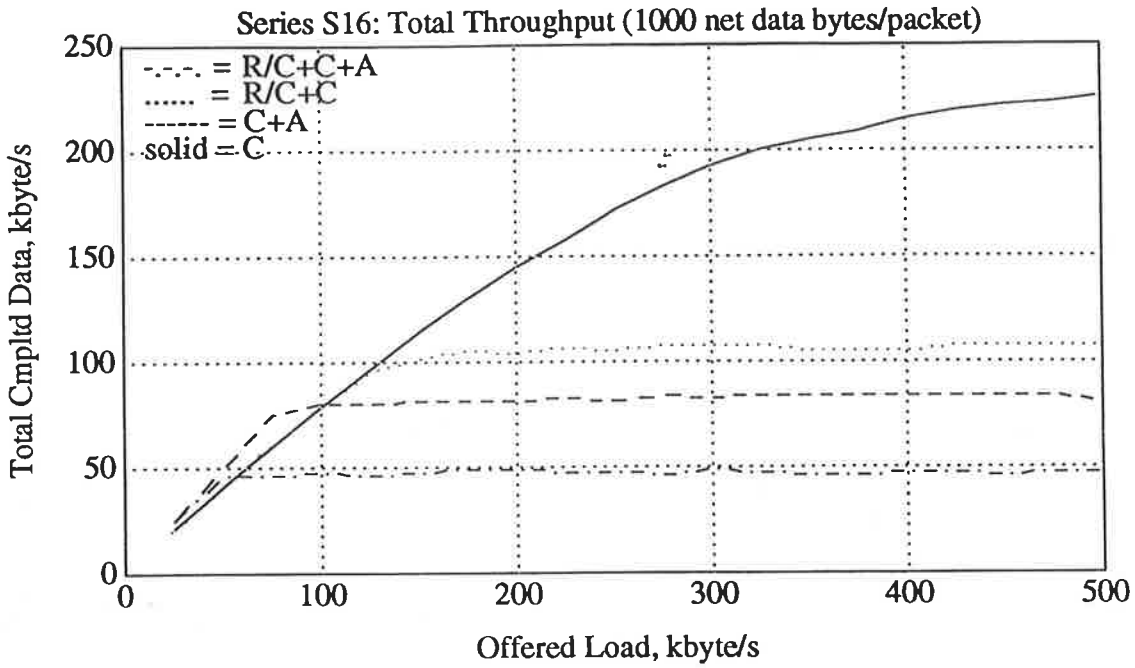


Figure 3. Throughput For Heavy Interference Case.



4. Source Queuing

RFMACSIM may be run with or without a source queue, in which blocks are held until the node's transmitter is free. The presence of the source queue strongly affects the simulation results. Figures 4 through 7 show the effect of source queuing on the acceptance rate and throughput curves. Series S10 (no queue) and S11 (queue length = 10) differed only in the queue length.

Without a source queue, the acceptance rate falls off at a steady rate starting at small offered loads. With a source queue, the acceptance rate curve has a definite "knee," below which the acceptance rate is essentially 100%. The location of this knee varies with the network geometry, protocol and other factors. The differences in the acceptance rate are clearly reflected in the shape and level of the throughput curves. In general, queuing leads to significantly higher throughput levels, especially at moderate offered loads below the channel capacity.

Figure 4. Acceptance Rate Without Source Queue.

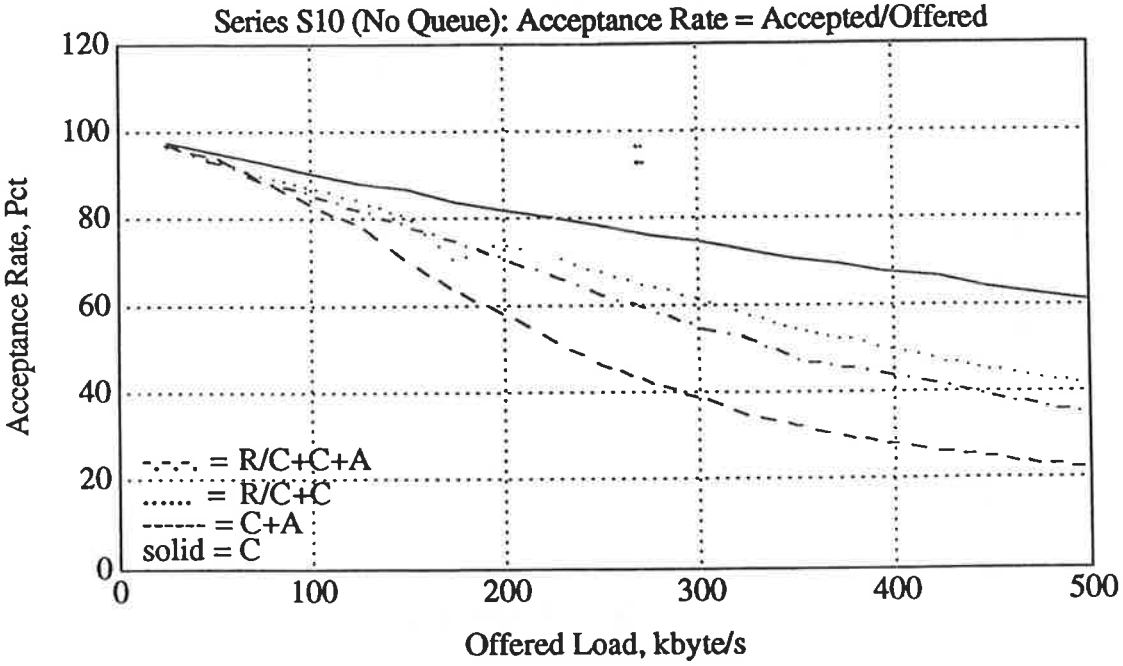


Figure 5. Acceptance Rate With Source Queue.

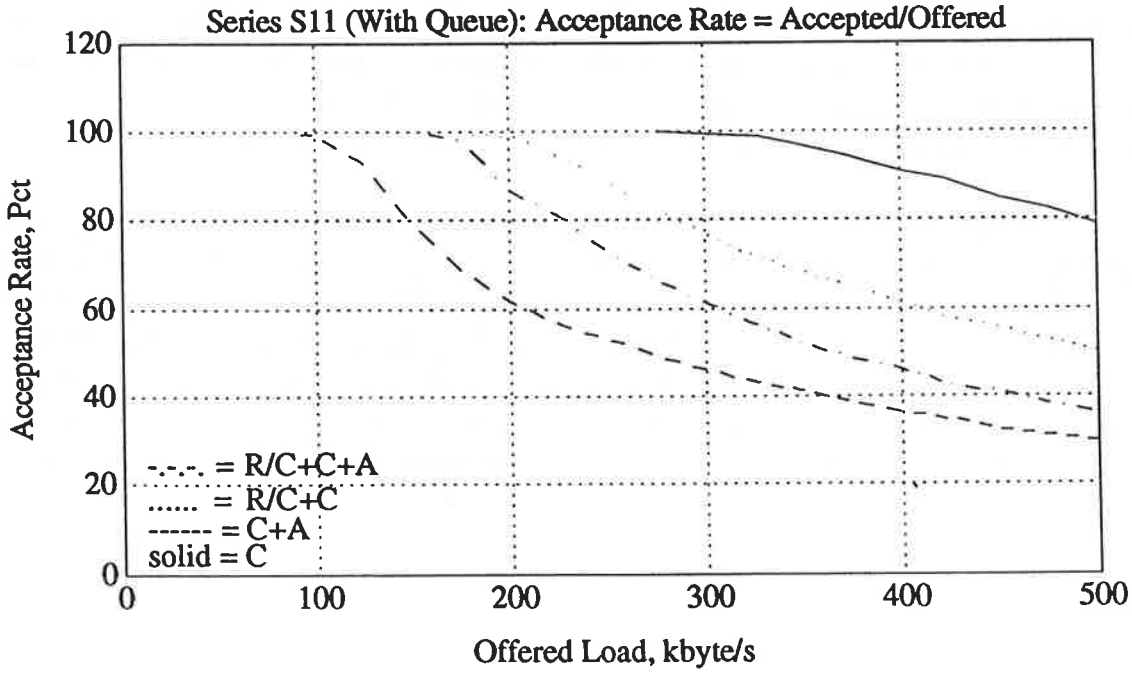


Figure 6. Throughput Without Source Queue.

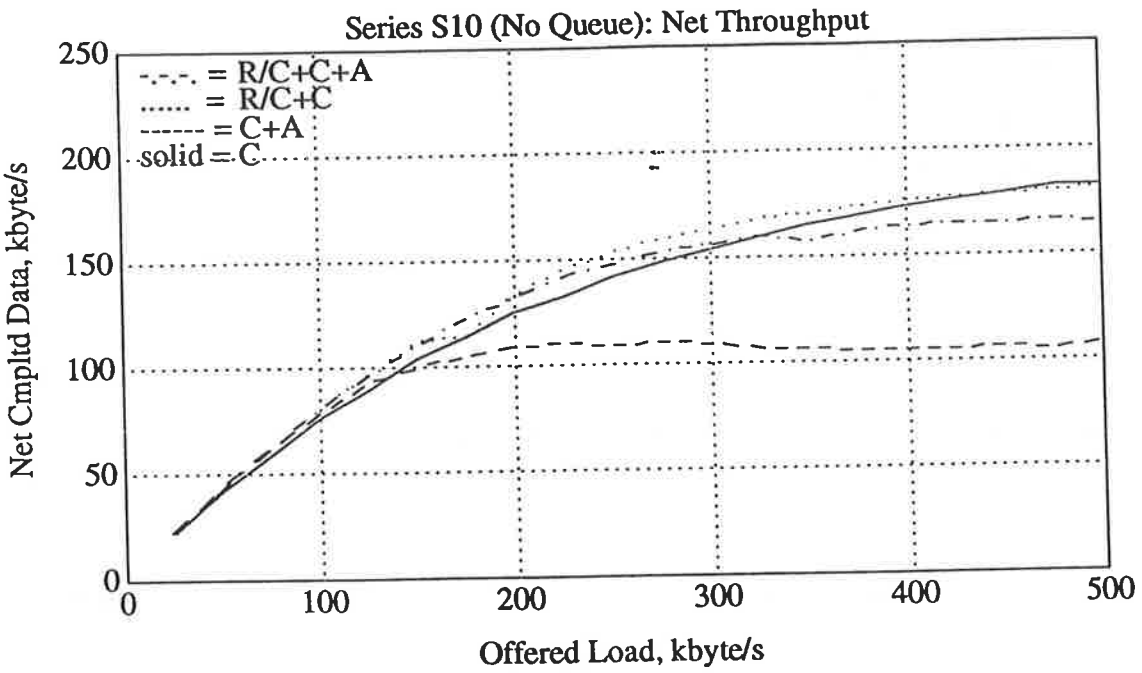


Figure 7. Throughput With Source Queue.

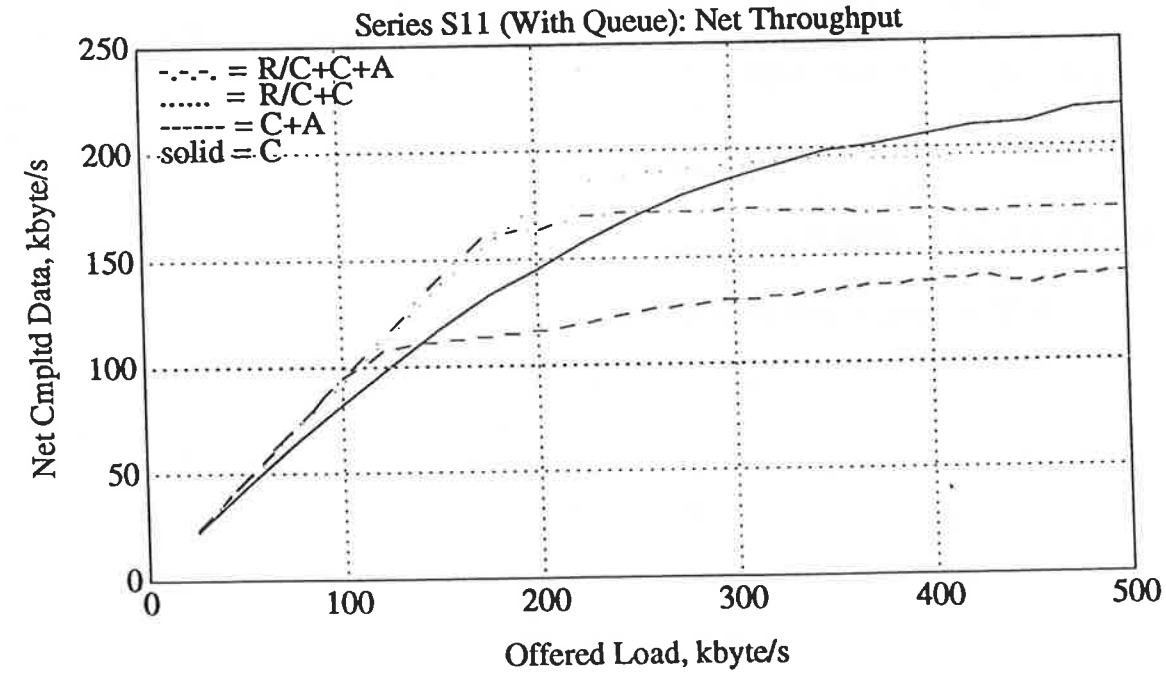
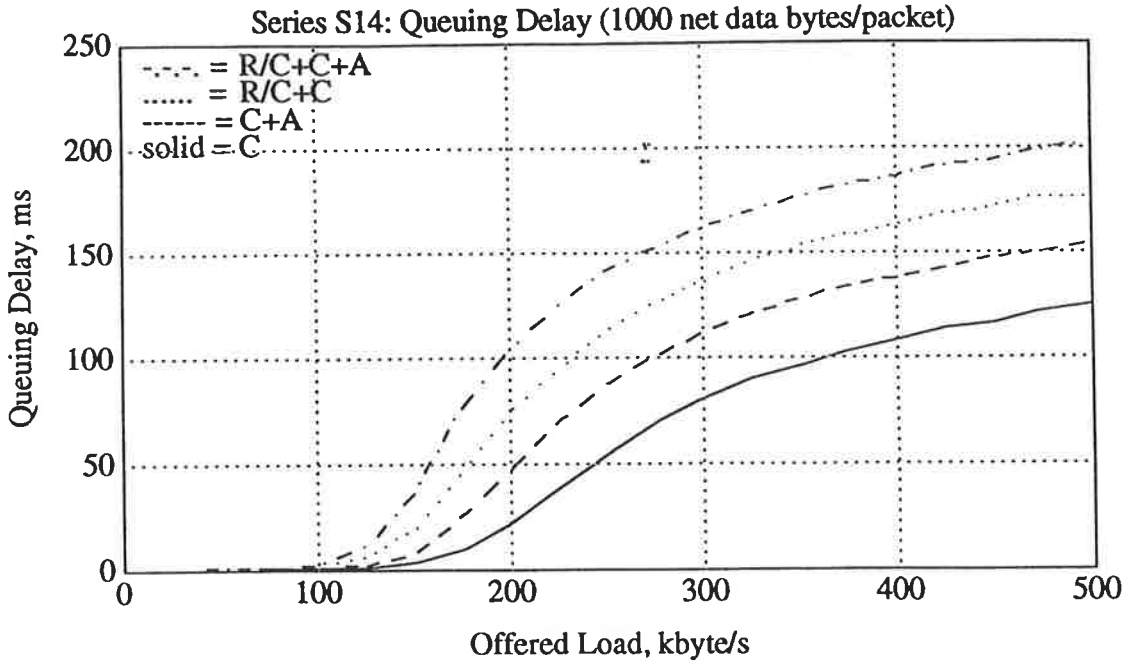


Figure 12. Queuing Delay For Normal Case.



8. Silenced CTS Problem

RFMACSIM generates a comprehensive set of counts describing the reasons for non-completion of protocol cycles. A review of these counts for several R/C+C+A simulations show that 15% to 25% of RTS requests go unanswered because the intended data destination node (DDN) is observing a prior off-the-air reservation. We call this behavior the "silenced CTS problem" (SCP). The remaining 75% to 85% of unanswered RTS requests are due to RTS or CTS packet collisions.

SCP comes about as follows:

The data source node (DSN) A transmits an RTS, and B correctly receives it, so that B goes off the air (OTH) for the intended duration of A's entire protocol sequence. Assume that B is not A's intended DDN, but just a bystander. A third node C does not correctly receive A's RTS.

A little later, C transmits an RTS to B as its intended DDN, and B correctly receives it. But, because B is still observing A's original OTH reservation, B cannot send a CTS to C. C will time out and may retry the RTS several times before B's OTH period lapses.

Figure 8. Throughput vs. Receiver Sensitivity For Hidden Nodes Case.

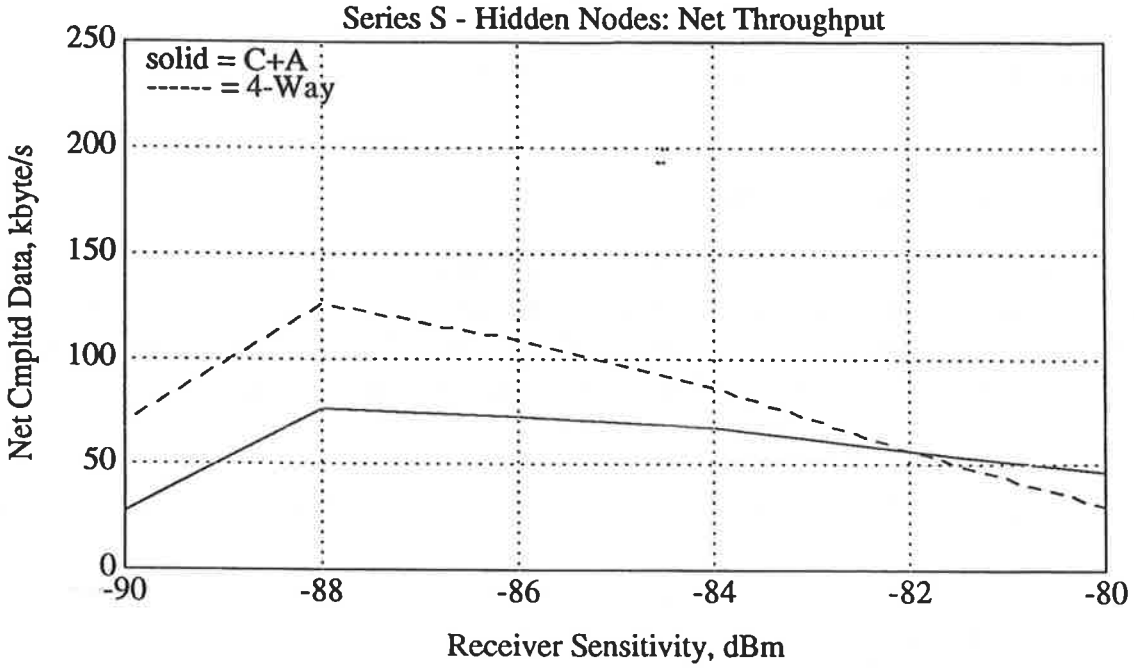
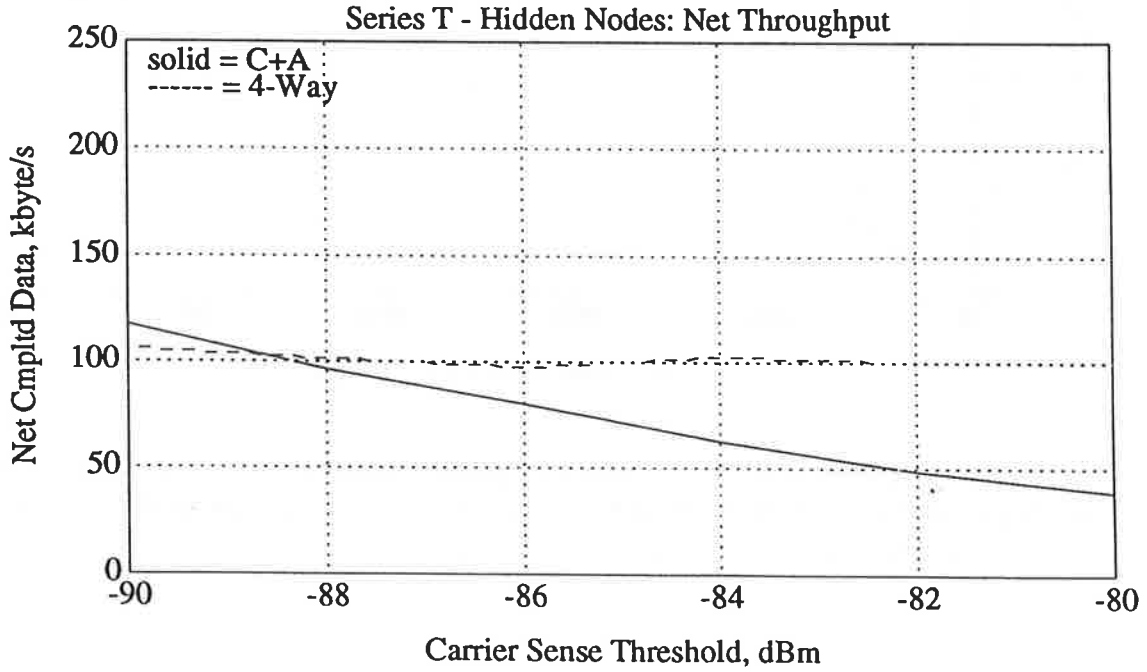


Figure 9. Throughput vs. Carrier Sense Threshold For Hidden Nodes Case.

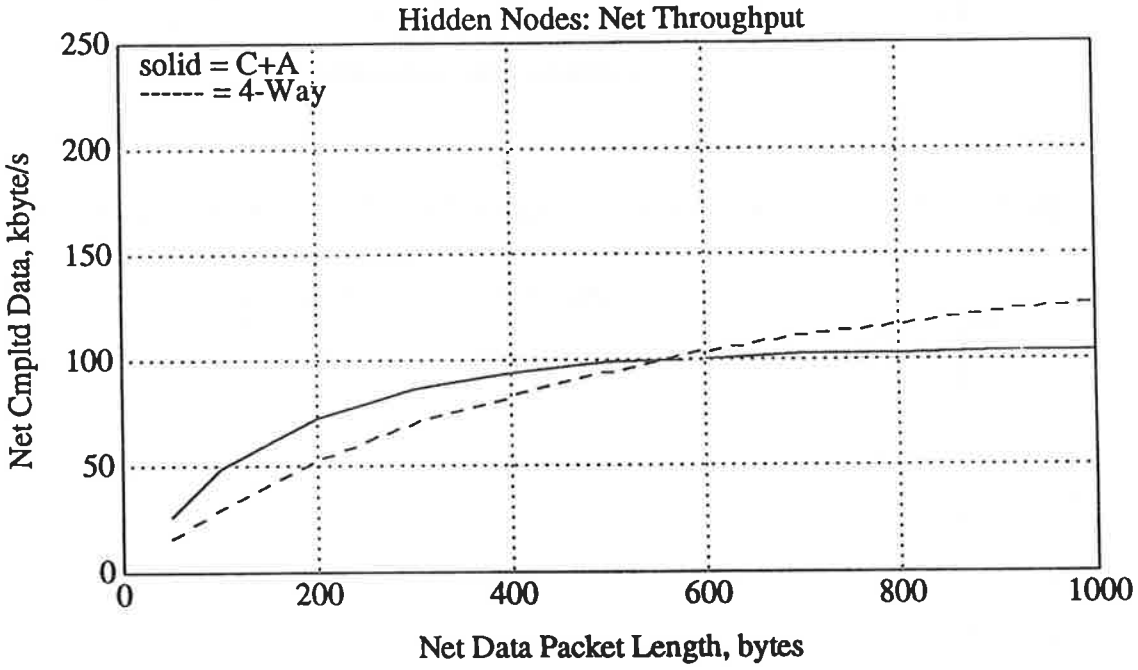


from each source node. As the sensitivity drops below a threshold level, an additional pair of nodes comes within range, and their inclusion as possible destinations leads to increased interference level. Since CST does not affect the simulated traffic patterns, the CST curves do not show the same kink.

6. Data Packet Length

A similar network throughput crossover occurs in comparisons of the C+A and R/C+C+A protocols as the data packet length is varied. As shown in Figure 10 (for the hidden nodes geometry), C+A yields higher throughput with shorter packets, while R/C+C+A performs best with longer packets.

Figure 10. Throughput vs. Data Packet Length For Hidden Nodes Case.



The cross-over point occurs at a smaller packet length in the heavy interference geometry than in the hidden nodes geometry. As noted above, the crossover point is also influenced by the RF parameters.

7. Transfer Delay

For some network environments, long packet transmission delays may be unacceptable to the network operating system, even though the wireless network throughput is otherwise satisfactory. Although RFMACSIM produces detailed delay statistics by delay type, including both average delays and delay histograms, our initial simulation work gave only limited attention to the delay results.

The simulator recognizes two major delay types: the queuing delay from the time of acceptance to the time the request is removed from the queue, and the transfer delay, from queue removal to protocol completion. Delay statistics are computed only for those requests that complete the entire protocol cycle. The longer queuing delay is heavily influenced by the maximum queue length (a simulator parameter). Thus, the transfer delay is a more appropriate measure of protocol performance. Figures 11 and 12 show sample average transfer and queuing delays for a normal network geometry.

In nearly all simulation results produced to date, protocols with larger throughputs have shown lower transfer delays. Thus, the relative protocol performance may be obtained from either the throughput or transfer delay graphs.

Figure 11. Transfer Delay For Normal Case.

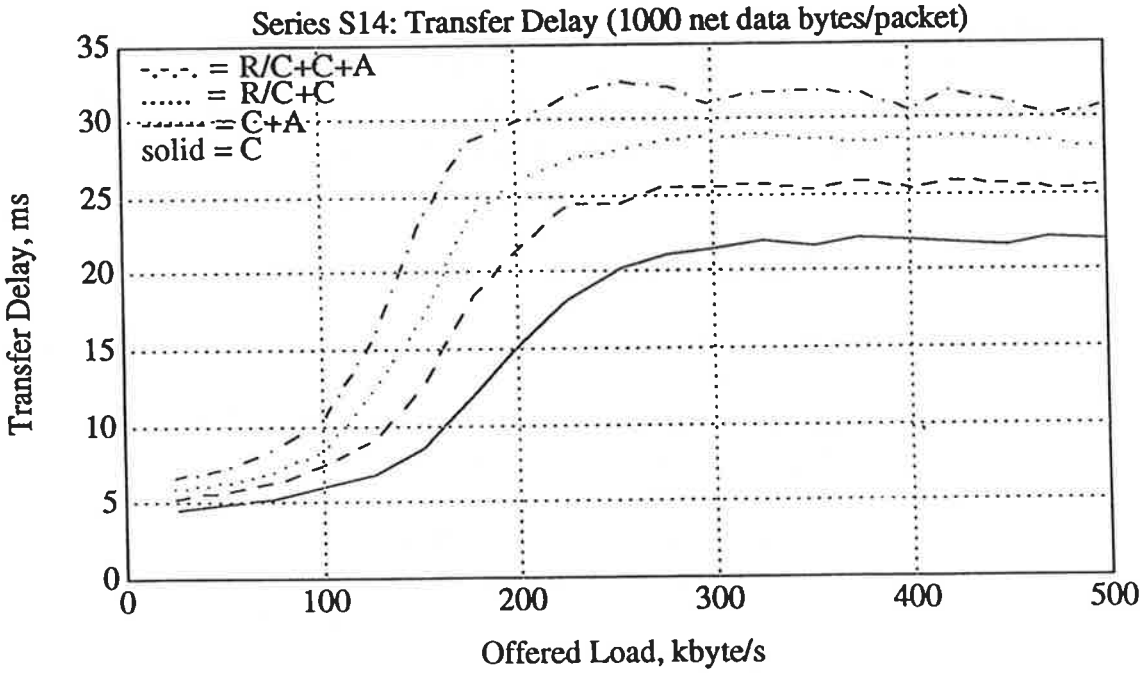
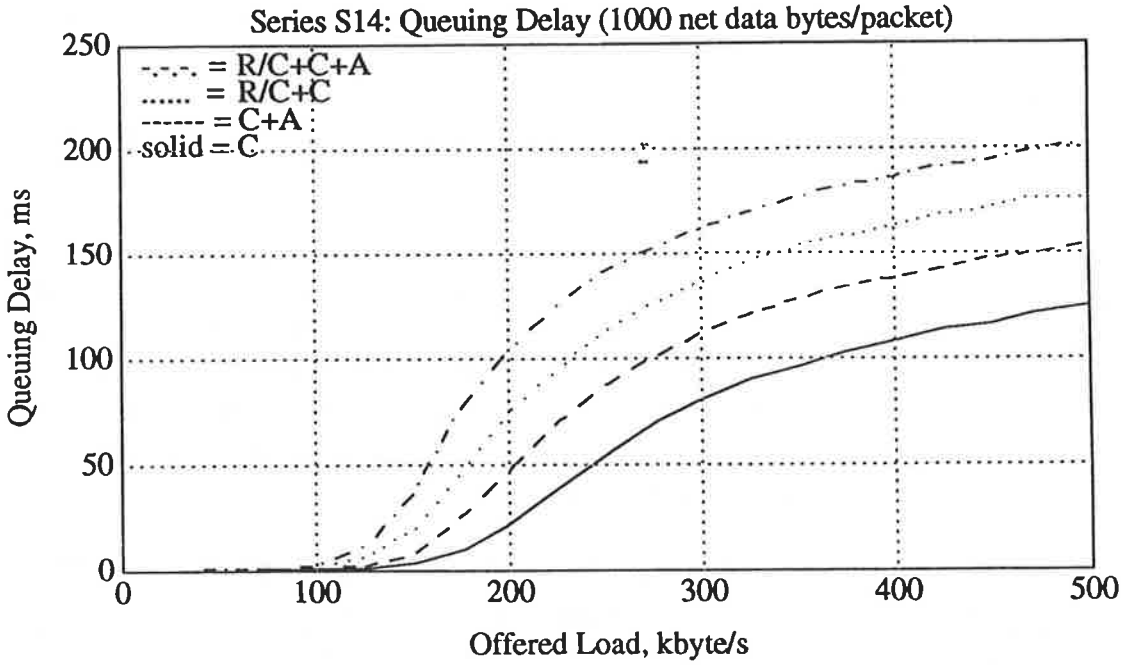


Figure 12. Queuing Delay For Normal Case.



8. Silenced CTS Problem

RFMACSIM generates a comprehensive set of counts describing the reasons for non-completion of protocol cycles. A review of these counts for several R/C+C+A simulations show that 15% to 25% of RTS requests go unanswered because the intended data destination node (DDN) is observing a prior off-the-air reservation. We call this behavior the "silenced CTS problem" (SCP). The remaining 75% to 85% of unanswered RTS requests are due to RTS or CTS packet collisions.

SCP comes about as follows:

The data source node (DSN) A transmits an RTS, and B correctly receives it, so that B goes off the air (OTH) for the intended duration of A's entire protocol sequence. Assume that B is not A's intended DDN, but just a bystander. A third node C does not correctly receive A's RTS.

A little later, C transmits an RTS to B as its intended DDN, and B correctly receives it. But, because B is still observing A's original OTH reservation, B cannot send a CTS to C. C will time out and may retry the RTS several times before B's OTH period lapses.

Similarly, SCP also occurs when B receives a CTS that C does not receive.

Although SCP resembles the hidden node problem (HNP), it can occur in a wider variety of settings. The key distinction between SCP and HNP is that an RTS or CTS packet may not be correctly received by C in many situations where C *is* able to sense A's carrier. Thus, SCP arises both from the capture effect (minimum signal to interference ratio) and from the RF propagation effects that give rise to the HNP.

9. Summary

A MAC protocol simulator as rich as RFMACSIM provides ample opportunity to explore simulated wireless LAN performance under a wide variety of assumptions. Our limited experience with RFMACSIM has taught us to avoid making categorical statements that protocol X "is better" than protocol Y. Relative protocol performance varies widely with the underlying assumptions about network geometry, RF parameters, data packet length, and other factors.

Having made this disclaimer, I will venture to state a few simple conclusions that are supported by our initial simulations:

- An ACK response (whether or not combined with RTS/CTS) is necessary to maintain high completion rates in environments with heavy interference.
- In order to keep completion rates near 100%, the no-ACK retry limit should be set as high as possible, subject to a reasonable bound on the maximum transfer delay.
- Adding an RTS/CTS exchange to the basic C+A protocol can improve throughput under some conditions (e.g., long packets), but may decrease throughput under other conditions. Some of the potential benefits of the RTS/CTS exchange are negated by the silenced CTS problem.
- The performance of both C+A and R/C+C+A is surprisingly sensitive to the values of certain RF parameters, which may vary significantly from radio to radio in a given production run.

References

[1] Carlos M. Puig. *RF MAC Simulator Documentation: RFMACSIM Version 0.32.* Apple Computer, Inc. July 29, 1993. Note: Version 0.32a is the most recent simulator version.

APPENDIX

Introduction to

"RF MAC Simulator Documentation: RFMACSIM Version 0.32"

July 29, 1993

The RF MAC Simulator (RFMACSIM) is intended to help wireless LAN developers evaluate the strengths and weaknesses of various proposed MAC protocols. The simulator includes a simple traffic generation model, a comprehensive indoor signal propagation model, and implementations of five MAC protocols: ALOHA, CSMA/CA, CSMA/CA + ACK, RTS/CTS + CSMA/CA, and RTS/CTS + CSMA/CA + ACK. The program's extensive set of simulation parameters permits simulation of a wide variety of scenarios.

Much of the implementation of the MAC protocols in RFMACSIM follows the suggestions of Wim Diepstraten of NCR Corporation. Mr. Diepstraten's submissions [1, 2, and 3] to the IEEE 802.11 committee served as the primary references for our development effort. He also provided Apple with the source code for NCR's simulator, and he helpfully clarified some aspects of NCR's simulation approach at a recent meeting.

RFMACSIM's underlying simulation approach differs considerably from that used in the NCR model. RFMACSIM includes a rewritten version of the "smpl" discrete simulation kernel presented by M. H. MacDougall of Apple Computer in [4]. In addition, the program uses the random number generators from MacDougall's text. Although the source code for the simulator kernel in RFMACSIM bears little surface resemblance to smpl, both take the same basic approach: Events are inserted in time and priority order on a linked list, and they are removed from the list and executed one at a time. This approach allows great flexibility in developing a simulation model, and can readily accommodate multiple events scheduled asynchronously for the same facility. Because no simulated activity takes place between events, the program's execution time is roughly proportional to the total number of events, rather than to the total simulated time.

The simulation program was developed in ANSI C to permit easy installation and efficient execution on a variety of hardware platforms. All program input and output is performed through standard text files. A separate application program must be used to generate graphical results. For the test runs

conducted so far at Apple, we have used MATLAB (Macintosh version) to easily produce a variety of graphs from the RFMACSIM results file.

Development of RFMACSIM began in late May 1993. The first fully operational version (v0.2) was completed one month later. The current version (v0.32) includes additional enhancements and bug fixes. Accordingly, the program has undergone limited testing and is likely to contain some errors. The current model also suffers from some design limitations that will be removed as time and resources allow. Possible enhancements include fading model improvements, more realistic traffic generation models, and the capability to simulate more than one network at a time. Comments, criticisms, and suggestions will be greatly appreciated.

This report is oriented toward those who wish to use and extend RFMACSIM. It contains little discussion of the important issues underlying indoor propagation and wireless MAC protocols. It contains only one sample set of simulation results, to help users verify the program's proper operation on their computers. Those intending to run, but not modify, the program should read sections 2 and 3, as well as the Appendices. Those intending to modify RFMACSIM should read the entire report side-by-side with the program's source code, which contains extensive internal comments.

References from "RF MAC Simulator Documentation"

[1] Diepstraten, Wim. "Wireless Network Performance Modeling Approach." Doc. IEEE P802.11-92/26. February 1992.

[2] Diepstraten, Wim. "A Wireless MAC Protocol Comparison." Doc. IEEE P802.11-92/51. May 1992.

[3] Diepstraten, Wim. "A Distributed Access Protocol Proposal Supporting Time Bounded Services." Doc. IEEE P802.11-93/70. May 1993.

[4] MacDougall, M. H. *Simulating Computer Systems*. Cambridge: MIT Press, 1987.