ABSTRACT
A formal specification and verification of a wireless LAN protocol is given, and a simulation of the protocol’s performance is also described. The verification is a partial verification in that it does not include a general verification of all possible configurations of the protocol; however, it does cover the expected usage of the protocol, which is for a small number of users sharing a cell. The simulation provides further insight into the protocol, especially concerning its performance. Based on the verification and simulation, some suggestions are made which can improve the protocol’s performance.

I. INTRODUCTION
Wireless communications in various forms has been the subject of much attention and research in recent years. Wireless communications include the broadcast radio medium; terrestrial microwave; satellite microwave in all its forms; and the International Amateur Radio Network, or “Ham Radio”. Additionally, wireless communications includes acoustical (sound) communications, used by submarines for many years, and also for data communications, at least experimentally; and optical (visible) light communications, which are still used by maritime vessels. More recent forms of wireless communications includes wireless telephones; cellular telephones; air phones, which enable the passenger aboard an airliner to connect to the long distance telephone system; and wireless computer networks. Pagers and remote control devices might also be included, though these are very “low bandwidth” communications devices.

In this paper we discuss a protocol suggested for wireless computer networks, known as MACAW. This protocol was suggested as an improved means to enable computers in a local area to communicate with each other. The protocol is intended for a single channel packet radio network. This means that all stations use the same channel (frequency), so that should more than one transmit at a time the signals may interfere.

Next we briefly describe MACAW. In section 2, the formal specification is summarized. Section 3 discusses the verification of the protocol. Further study of the protocol was carried out through a series of simulations; this is described in section 4. As a result of this work what we have learned through the formal specification process, the verification, and the simulation -- we are able to make some suggestions as to how it can be improved, and to give a fairly accurate indication of how much the improvements will help performance.

MACAW -- Multiple Access, Collision Avoidance, Wireless -- was designed as an improvement upon an earlier protocol, called MACA [Kar]. MACAW is based on the same principle as is the cellular radio network for automobiles. A set of base stations, which are all connected by a wired system, communicate via radio signals with the user stations. The difference is that with MACAW, the base stations are connected by a local area network -- such as CSMA/CD -- rather than a telephone type network covering a large area. The area around each base station is called a cell; and the users communicate with the nearest base station. Another difference is in the cell size; here we are using cells which are only a few meters in diameter, rather than miles across. A third difference is that the stations are not mobile, as are autos. The signals are relatively weak, so that they should not penetrate beyond the “cell,” or its local area. However it is possible that some interference between adjacent cells may occur. Other protocols for wireless and broadcast networks exist, and have been in use for years; these are discussed in the references.

MACA (medium access/collision avoidance) [Kar], MACAW’s predecessor, used a sequence of three basic messages: request-to-send, clear-to-send, and DATA. MACA used binary exponential backoff algorithm to resolve collisions, the same type used in CSMA/CD. Studies showed that the MACA protocol was not fair to all stations on the network, and the throughput was not very high under conditions of imperfect channels [BD].

MACAW (MACA-wireless) improved upon MACA by adding to this message sequence. The request-to-send (RTS) message from the sending station is followed by the clear-to-send (CTS) from the receiver, as with MACA; but the DATA is preceded by a short data-send (DS) packet, which serves to alert the nearby stations that a data packet is about to be transmitted, reducing the probability of a collision. This is followed by an ACK packet from the receiver, through which reliability is enhanced. This means that MACAW is able to provide reliability at the MAC layer, where MACA could not.

MACAW also uses a different backoff algorithm than MACA. MACAW’s developers felt that the MACA
backoff algorithm oscillated too rapidly. Instead the value of the backoff counter is increased by a factor of 1.5 (rather than 2) for each collision, and decreased by 1 for each success. This provided a gentler fluctuation in the backoff counter.

In this paper we show that MACAW is a reasonable protocol for its intended purpose, however that it can be slightly refined and improved. Our contributions include the formalizing of the protocol specification, its verification, additional knowledge of its performance, and suggested improvements.

II. SPECIFICATION OF MACAW
MACAW is formally specified and analyzed using a model called systems of communicating machines. This model, originally defined by Lundy and Miller [LM] has been used to analyze several communications protocols. Each machine is defined as a finite state machine with variables. Communication between machines is done through the use of shared variables. Local variables are used to maintain the state information for each machine, and a predicate-action table serves to unify the state machines and the variables. The goal of formal specification is to determine if the protocol is free from errors, such as deadlock or unexecutable transitions, and if it possesses “liveness,” which is defined as the ability to make positive progress. An additional benefit is identification of ambiguities in the protocol.

A formal specification of MACAW was developed in [AI], which includes a set of local variables, a set of shared variables, used for communication, a finite state machine, and a predicate-action table.

The shared variables are channel.type, channel.DA, esn, channel_my_backoff, channel_local_backoff, and channel_retry_amount. These are used for communication between the nodes, and the names are descriptive of their uses.

The local variables are inbuffer, outbuffer, backoff, my_backoff, local_backoff. Again, the names of the variables are descriptive of their uses.

The finite state machine diagram has 10 major states. These states are named IDLE, CONTEND, XMIT, WFCTS, WFACK, WFDS, QUIET, WFDATA, WFCONTEND, CONTEND2.

The names of the states are descriptive; in IDLE, the node has nothing to transmit. In CONTEND, the station has a message to transmit and is trying to get the channel; XMIT, is sending; “WF” signifies “Waiting for,” as in WFACK, “waiting for acknowledgment.” The initials CTS signify “clear to send.”

The predicate-action table unifies the finite-state diagram with the local and shared variables. Each transition in the diagram has an entry in the table, and for each entry there is an action and an enabling predicate. The full specification is given in [AI].

III. VERIFICATION OF MACAW
A major benefit of formal specification of protocols or computer programs is the potential for automated generation and analysis. This automated analysis was conducted using the program developed in [Bu]. The analysis is done by constructing a directed graph of states and the transitions between states.

Three basic configurations were analyzed with the automated analysis. The number of machines was varied by using three input files with two, three and four machines. Though it is possible to have more than four machines in a single cell, this is unusual; a cell is only a few meters in diameter. So, while our verification does not cover all cases, it does cover the majority of practical cases.

All of the possible specified transitions of MACAW were not always executed by all machines; in fact, for some cases, all transitions were not executed by any machines. At first glance this appeared as a protocol error -- an unexecutable transition -- but in fact was not. It was simply a case in which the transition in question was not needed. For example, in the two machine analysis, none of the quiet state transitions were executed for the simple reason that there was no other machine causing interference.

With two machines: Machine 1 and machine 2 were run in the two machine analysis. Both began in the IDLE state and machine 1 had a message to send to machine 2. Once machine 2 received the DATA message from machine 1 it sent a DATA message to machine 1. (Machine 2's action upon receiving a DATA message included writing a DATA type message to its own output buffer). No deadlock occurred and the total number of system state generated was 69. The corresponding global state analysis generated 275 states. One artificial deadlock occurred in the global state analysis upon receipt by machine 2 of the ACK message from machine 1. The deadlock happened as a result of clearing the output buffers when an ACK message was received. Essentially, both machines received their respective DATA messages and no more data remained to be transferred, thus causing a deadlock.

Simulations with 3 and 4 machines, which generated many more states, are reported in [AI].

IV. SIMULATION AND ANALYSIS OF MACAW
The purpose of the simulation is to determine the behavior of the protocol under different operational conditions. Therefore, operational conditions are first determined as test cases and simulations are performed with these test cases. The results are analyzed and some suggestions to improve the protocol are given.

The simulation model of the protocol was built using OPNET 2.4c from MIL3, Inc. The OPNET design
environment greatly eased the simulation process.

A. Test Cases. The behaviors that we are interested in are the utilization of the channel and packet mean queue delay time. The usual performance measure for a protocol is network utilization, but this does not consider users of the network. Therefore, mean queue delay time is also considered as a performance measure of the protocol. These measures are related to the communication channel load. The load of the channel depends on the number of the nodes that are sharing the same transmission media (channel) and their (internal) packet arrival rates. Besides these, the performance of the radio networks is greatly affected by hidden-exposed node/cell conditions [Kar].

With these considerations in mind, the test cases are separated into two groups known as “load cases” and “hidden-exposed node/cell cases.” In the load cases, it is also determined whether the protocol acts in favor of any specific communication topology. For each test case, communication scenarios are determined and, for each scenario, test cases are generated with different numbers of nodes and packet arrival rates.

With each load case, networks of up to four transmitting nodes are built and each of these cases is tested from 10% to 100% of the total offered channel loads. Having up to four transmitting nodes for each load case provided enough information about the behavior of the protocol under these conditions. Sample load case scenarios are shown in Figure 1. The load case topologies of interest are as follows:

(a) reporting nodes with two, three and four transmitting nodes; (b) one-way communicating nodes with two, three and four transmitting nodes; (c) cross communicating nodes with one and two pairs mutually communicating nodes; (d) ring communication with three and four nodes.

These cases are also tested from 10% to 100% of the total offered overlapping channel loads. An overlapping channel is the intersection of transmission ranges of two (or more) non-communicating nodes where one’s transmission interferes with the other. A brief explanation of each case follows:

Hidden node: While A is sending to B, C attempts to send to B also. In this scenario A and C are not aware of each other but affected from each other’s transmission, so they are hidden from each other. Exposed node: While B is sending to C, A attempts to send to B. Here B is aware of A, thus B is exposed to A’s transmissions. Hidden and exposed nodes: While A and B are mutually exchanging data, C attempts to send to B. Here A and C are hidden from each other and B is exposed to both A and C. Hidden Cell: B and C are in different cells but they can hear each other. Node A is not aware of the communication going on in the adjacent cell since it cannot hear anything from that cell, it is hidden from A, the same is true for node D. Exposed Cell: B and C are in different cells but each can hear the other’s transmissions while they are communicating with other nodes in their cells (A and D respectively). So they are exposed to each other’s transmissions. Hidden and Exposed Cells: Node A does not hear any transmissions from the other cell while node C is hearing node B. So the adjacent cell is hidden for A and exposed for C. All in one: Heavy communication in both cells.

B. Performance Measure Criteria. As mentioned above, the performance measure also includes the mean delay time in addition to utilization. The resultant formula is:

\[
\text{Total Performance} = \frac{\text{utilization}}{\text{mean-queue-delay}}
\]

According to this formulation, if two utilizations are equal then the one with lesser mean delay time will have a better performance measure. For example, suppose we have measured an utilization of 54% with a mean delay time of 9 sec and for another case 54% utilization with a mean delay of 3 sec. Just looking at the utilizations does not say anything about which one is better. But when delay times are considered, the first case will have a performance measure of 0.54/9=0.06 while the second case will have a performance measure of 0.54/3=0.18. This clearly shows that the second case is better.

A performance measure should be independent of transmission speed of the network in order to be used to compare different protocols. Thus, delay time is normalized to data packet length. This means we can compare the performance of the protocol for networks with different transmission speeds.

C. Test Results. For all cases the Poisson distribution function is used for the packet arrival distribution. The first test case, our base case, is chosen to determine the
maximum performance of the node that is running MACAW protocol for a given arrival rate. This test is conducted with a network of one transmitter and one receiver, and utilization is calculated. Data mean queue delay times are collected by OPNET’s built-in utilities. OPNET calculates the delay times of the packets that arrive to the queue during one unit of time and uses its mean (middle) value as a mean delay time of this time period. The simulation result of this base case and the other cases are shown in detailed tables in [Or].

As seen from the table, the performance decreases rapidly beyond 80% offered channel loads due the high mean delay times. Maximum utilization observed for the protocol with given packet sizes (30B for control packets, 520B for data packets), packet arrival distribution function and backoff limits is 66%. These results will be a base case to use in comparison with test results.

Load Case Results For the load cases, channel utilization almost never exceeds 50% for all simulations (the maximum utilization observed for the protocol was 66% with current parameters). The exceptions are the base case and “one pair mutually communicating” network case. In the base case there is no interfering node against the transmitting node, so we should get “the most” from the protocol. The reason that one pair mutually communicating network has high utilization and low backoff values than the other simulation cases is it has a slower rate of backoff increase for its remote backoff values.

For high loads, one-way communicating nodes cases always violate the fairness of channel access. The backoff algorithm dictates that each node keeps track of the congestion of the channel between itself and the neighboring nodes via backoff values. When a node attempts to communicate with one of its neighboring nodes, it uses the associated backoff value. These backoff values are mutually exclusive. In a one way communicating network scenario when a “lucky” node reaches the channel quicker than the others for a couple of consecutive times it gets and keeps the channel by knowing other nodes have very high backoff values.

Reporting and ring cases have similar results. They both have fair channel access during the simulations. But both have high backoff values. To have high backoff value causes larger delay times on average therefore reporting and ring cases have lower performance measures.

The best performance measures observed during the simulations are around 20% channel load conditions for three and four transmitting nodes cases. As the number of transmitting nodes in the network increases, the effect of having high backoff values decreases (for low loads), because when the channel is free, the chance that there is a node ready to transmit is high. So idle times of the channel are used and total performance measure of the network increases, since per node utilization and data mean queue delay times does not change.

Hidden and Exposed Node/Cell Case Results In the hidden node case, due to the rapid increase of the backoff value of node B, performance is poor above 40% of overlapping channel loads (load over node B). Channel access is fair, because both transmitters are using the same backoff value as in reporting nodes case of load cases.

In the exposed node case performance is fairly high compared to the hidden node case. However, above 80% of offered overlapping channel loads, the behavior of the network becomes similar to the one way communicating node cases of the load cases. The “lucky” node gets the channel and keeps it.

In the hidden and exposed node case, the hidden node behavior is dominant. Due to the high delay times, the performance decreases dramatically above 40% of overlapping channel loads.

In the hidden cell case channel access of transmitter nodes (node A and D) are always fair because the congestion levels at the receivers are similar. But due to the congestion at the receiver side data packet delay times in the transmitters are high so performance is low for cell loads above 25%. Its behavior is the same as the hidden node case.

Modified Protocol Simulations During the simulations it was observed that the backoff increase rate is one of the key factors that affects the performance of the MACAW protocol under moderate and high loads (40% and above). To further investigate, the same set of simulations were run with a lower backoff increase rate. As a result, for load cases with a small number of transmitting nodes, significant performance gains were observed. These gains are due to the lower mean delay times rather than increased utilization. But as the number of the transmitting nodes increased, the performance gain achieved by this modification degraded. No significant performance change was observed in hidden-exposed node/cell cases.

Another factor that affects the performance of the protocol is the nodes’ inability to determine whether the channel is busy. According to the MACAW protocol, nodes behave purely in accordance with the received packet information. The keyword here is “received.” The MAC layer that runs the MACAW protocol does not know that it is receiving a packet until the entire packet has been transmitted. The probability of collision of two transmitting nodes is the probability of transmission of one node without recognizing (or before recognizing) that another node is transmitting. The transmission probability of a node when it assumes that the channel is available is based on the distribution function and its parameter value (in this case Poisson distribution with
arrival rate value). The “recognition” period in the original MACAW protocol is the time that it takes for control packet transmission plus propagation delay. Thus, we have a sample space of time_slot + propagation_delay for a transmission event that may cause a collision. When carrier sensing is introduced to the protocol, this sample space shrinks to one propagation_delay (assuming time_slot >> propagation_delay). So, carrier sensing will decrease the probability of collisions by a factor of time_slot/propagation_delay. This is also what is observed during the simulations of the MACAW protocol with carrier sensing capability. Significant performance measure increases observed for 50% and above common channel loads. The source of gain is offered load), no change was observed in the hidden cell case. In hidden cell case carrier sensing provides no help to the network to improve its performance, because transmitters are out of range from each other (the only time that a transmitter utilizes carrier sensing in this topology is to sense its receiver’s RRTS packet).

Another improvement achieved by carrier sensing is fairer channel access of the transmitting nodes (compared to the original network) in the “hidden-exposed cell” and “all-in-one” network topologies.

In “hidden-exposed cell” network node C cannot get the common channel as frequently as in the original protocol. Now C should wait B to send all its packet and act accordingly (node C must understand what B is saying”). This increases the performance of node A slightly, because backoff values are dominant for the performance and here there is a big difference between congestion levels of the receivers (especially for high loads).

In “all-in-one” case fairness is achieved between inner nodes. In the original protocol for high loads (80%), one of the inner nodes has much higher performance measure than the other. Now with carrier sensing both have similar performances. The performance difference between inner and outer nodes are still great because of the congestion levels at the receivers.

Simulation Summary. It was determined that the backoff algorithm has a significant importance for the performance of the protocol, where performance is defined in terms of channel utilization and transmitter mean delay times.

Backoff is the measure of the congestion at the location of the node. Distribution of this information throughout the network provides fair access to this node by other nodes since each node keeps track of the other node’s congestion information in the network. But this does not help in the situations like one way communicating nodes cases. Also some performance degradation was observed, which was caused by the rapid increase of the backoff increase rate for the small number of transmitting nodes cases. A relation between the backoff increase rate and the number of transmitting nodes was observed (but this needs to be investigated more thoroughly). So, instead of using a linear increase rate use of dynamic or exponential increase rates may give better results. The dynamic increase can bind to the number of transmitting nodes. For the fairness problem in one-way communicating nodes cases more fundamental changes should be made to the algorithm. That is, the backoff algorithm needs to be thoroughly analyzed.

It is also observed that the channel acquisition with carrier sensing is superior to the original MACAW protocol policy. This is quite significant for medium and high loads. This contradicts the reasoning of the original MACAW protocol specification which says “..carrier sensing is inappropriate” [BD]. This simulation is discussed in more detail in [Or], as well as the remainder of the simulations which are omitted here due to lack of space.

V. CONCLUSIONS
We described the specification, verification, and simulation of MACAW. The specification uncovered some ambiguities in the original specification. The verification showed the protocol to be free of deadlocks, for most practical cases. Simulations gave further confidence in the protocol and showed further possible improvements in performance. We showed that the backoff algorithm is of major importance to the performance of the protocol, and topology is also critical. The protocol can also be improved by adding carrier sensing. Simulations suggest that this could improve utilization under heavy load by about 5%.

REFERENCES