

# Performance of the HF Token Protocol

Eric E. Johnson, Gary Anaya, Zibin Tang, Manikanden Balakrishnan,  
Huiyan Zhang, and Srugun Sreepuram  
Klipsch School of Electrical and Computer Engineering<sup>1</sup>  
New Mexico State University

## ABSTRACT

*Token passing can provide efficient medium access control in heavily loaded networks. However, the management overhead required in forming and maintaining a ring of token-passing nodes is a potential liability for this protocol. In this paper, we present the results of both simulations and measurements of the HF Token Protocol in wireless LAN operation, and explore the range of applications that may make efficient use of token passing.*

## 1. INTRODUCTION

The ability of high frequency (HF) radio to communicate beyond line-of-sight range sometimes involves the use of very challenging channels. The robust waveforms required in such channels can necessitate long interleavers and extensive signal processing, resulting in end-to-end signaling delays (and turnaround times in interactive applications) on the order of seconds.

A study of the impact of such long turnaround times on the performance of media access control (MAC) protocols [1] found the following:

- Under light traffic loading, contention-based protocols such as the IEEE 802.11 (WiFi) DCF [2] offer lower latency than contention-free protocols such as TDMA. Under heavy traffic loads, however, use of a contention-based MAC protocol leads to severe congestion and degraded network throughput if turnaround times are long.
- TDMA provides efficient channel access control under heavy load, but requires network synchronization and management intervention to assign and re-assign slots to network members, and has relatively long latency under light load.
- Token passing [3] also requires some overhead, but its performance is attractive under both light and heavy loading (except in large, lightly loaded networks with long turnaround times, where its performance suffers relative to contention-based protocols).

The effect of long turnaround times when traffic is heavy is illustrated in Figure 1 (from [1]).

- **DCF** is the Distributed Coordination Function from IEEE 802.11 [2].
- **DCHF** is similar to DCF, but is optimized for heavy loading by incorporating a contention window backoff algorithm similar to MACAW [4], and by eliminating a carrier-sense interval (DIFS) that is of limited value in legacy HF radio networks.
- **TDMA** is a simple fixed-slot time division multiple access scheme.
- **Token** is the HF Token Protocol [3].

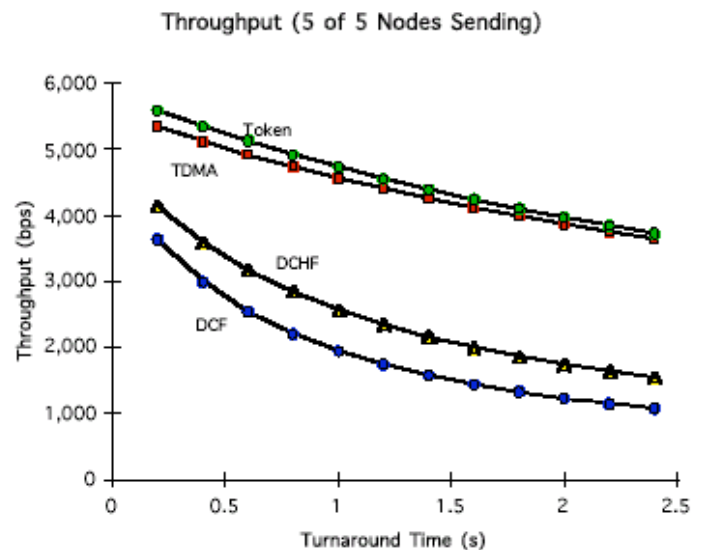


Figure 1: MAC Protocol Throughput–Heavy Load

All of these are steady-state performance estimates using analytical models; simulations of the same protocols *in steady state* corroborate these results [5]. However, the overhead required to set up a token-passing ring, and to maintain its smooth operation when connectivity changes, could negate the apparent performance advantage of token passing.

<sup>1</sup> This work supported by US Navy Space and Naval Warfare Systems Command, contract N660001-3287-2827LG

In this paper, we report the results of an investigation of such transient behavior of the HF Token Protocol (HFTP), using measurements of the initial implementation of HFTP by the US Navy SPAWAR Systems Center as well as simulation results. We begin with an overview of the HF Token Protocol, followed by a discussion of the simulation scenarios and results, and comparison with the measurements of the prototype implementation.

## 2. THE HF TOKEN PROTOCOL

A MAC protocol controls access to a channel that is shared among cooperating nodes. Token passing is a contention-free protocol because all of the nodes agree that only the node that currently holds a notional “token” is allowed to transmit. The cooperating nodes form a ring for the purposes of passing the token. In Figure 2 this token flow is symbolized by arrows connecting predecessor to successor nodes.

To promote fairness in channel access, a node can only hold the token a bounded time before it must pass the token to the next node in the ring. Furthermore, if a node has no traffic when it receives the token, it must pass the token immediately.

Token-passing protocols have been used in both “token ring” and “token bus” topologies. In both cases, the token is passed from node to node around a logical ring. However, in a token ring topology, data also flows only around the ring, forwarded as necessary through intervening nodes between the source and destination. In a token bus topology, on the other hand, the shared channel operates in broadcast mode at the physical layer, and data can be sent directly from source to destination. The HF Token Protocol (HFTP) is a token bus protocol, so each node can (nominally) send data directly to any or all other nodes.

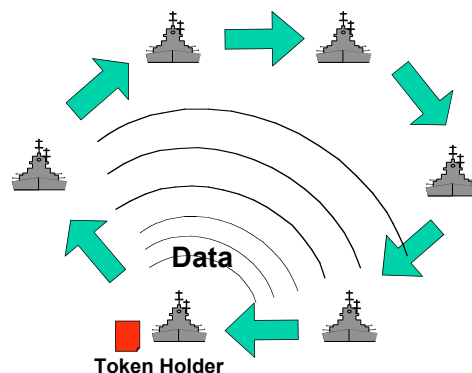


Figure 2: Token Passing in a Maritime Wireless LAN

The operation and performance of HFTP in steady state have been presented elsewhere [1, 3]; the focus of this paper is the transient behavior of HFTP, in particular the time required to form a ring where none exists. Here is a brief summary of the mechanism for linking disconnected nodes into a ring:

- Figure 3a shows a token ring in operation, consisting of nodes A, C, D, E, and F. Node B is within range of at least nodes A and C but is not yet part of the ring. To ensure contention-free operation, B cannot transmit on the shared channel until it receives the token, and must wait to be invited to join. While B is in this “Floating” state, it monitors the channel and records the network members from which it receives packets. These nodes are possible predecessors and successors when it joins the ring.
- The mechanism for adding nodes to a ring is periodic invitations from each ring member for floating-state nodes to become the soliciting node’s successor in the ring. The Solicit Successor packet names the current successor of the soliciting node. Prospective responders are only allowed to respond if they have received a

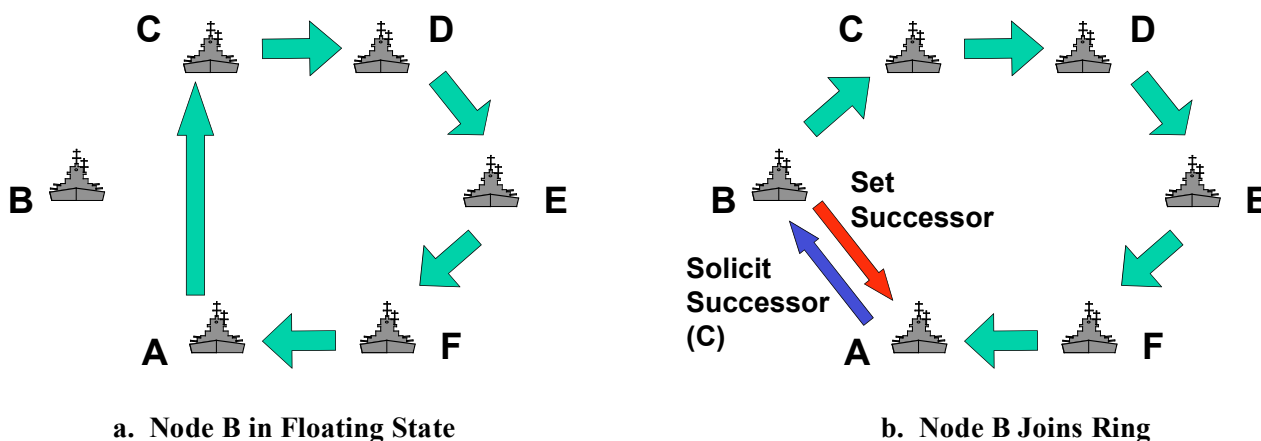


Figure 3: Mechanism for Joining an Existing Token Ring

packet from the named successor. This ensures that the ring is not broken when a new node is inserted.

- A Floating-state node (B) that receives a Solicit Successor packet and can reach the named current successor (C) returns a Set Successor packet in a randomly-chosen slot following the solicitation. (Collisions are possible at this point because more than one node may be in the Floating state.)
- The soliciting node (A) selects one of the responding nodes and send the token to its new successor (here B). The new node may then use the token or pass it immediately to its new successor (C).

The situation is slightly different when no ring exists. Nodes that have not overheard HFTP packets from any node enter a “Self Ring” state. In this state, they periodically broadcast Solicit Successor packets (which name themselves as the current successor). Any nodes that hear the solicitation respond as above, and a two-node ring is formed. From this point, additional nodes are added to the ring as previously described.

A third type of topology change occurs when a node leaves the ring. In this case, the departing node simply links its current predecessor to its current successor and departs.

Of the three types of HFTP ring topology changes, ring formation is the most complicated, because the opportunities for contention are the greatest. It is this ring formation operation that we will analyze in this paper.

### 3. SIMULATION SCENARIOS

The HFTP was implemented first in the NetSim simulation environment, then by the US Navy SPAWAR Systems Center in the BattleForce Email STANAG 5066 [6] stack. This initial application requires efficient operation in networks ranging up to at least 15 nodes, and must operate correctly even if some of the nodes are “hidden” from other nodes.

Simulations of a range of fully and partially connected topologies were used to explore the ring-forming behavior of HFTP. The simulation topologies are listed in Table 1, and a selection of them are diagrammed in Figure 4. In the figure, nodes shown as open circles are hidden from the bottom node or nodes in each topology, as described in the table.

A simple channel error model was employed in these simulations: a packet loss rate is specified for each simulation, and packets are randomly discarded in the channel with the specified probability.

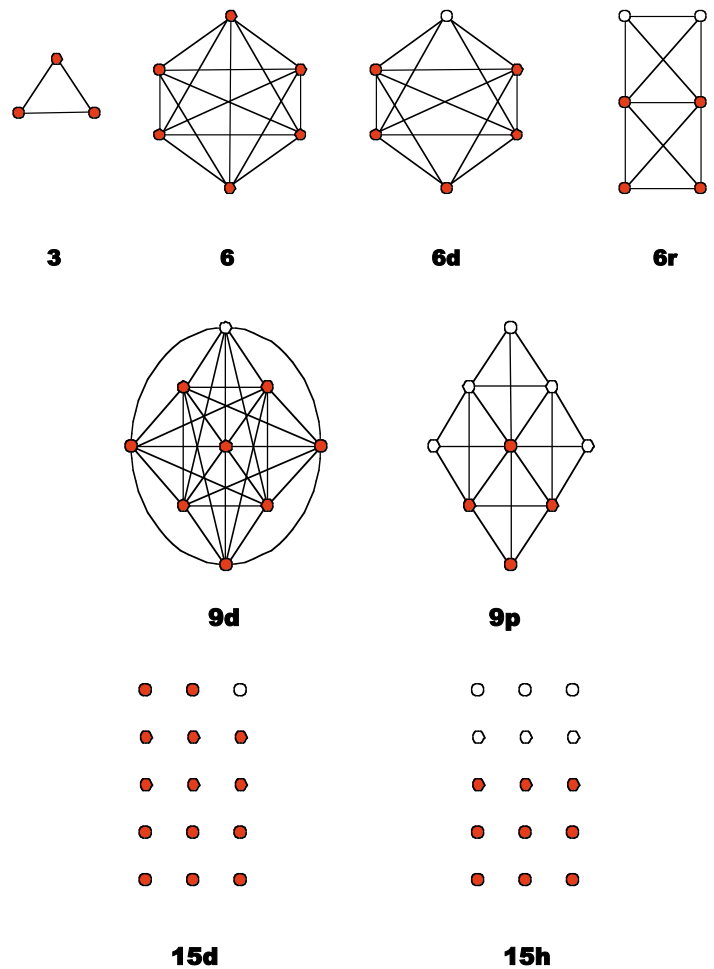


Figure 4: Network Topologies

The key operating parameters for the protocol in the simulations were as follows:

Turnaround time	2.24 s
Token pass time	5.28 s
Data rate	6400 bps
Solicitation slots	Adaptive

The number of slots for responses to solicitations used an adaptive algorithm that reduces contention during ring formation. Each solicitation includes the number of slots available for responses. When a node starts (or restarts) the protocol, it employs a large number of slots (e.g., 20) so that the potentially large number of responses is spread out. The number of slots is reduced for each succeeding solicitation, based on the number of responses received.

**Table 1: Simulated Network Topologies**

Topology	Nodes	Hidden Nodes	Description
3	3	–	Fully connected
6	6	–	Fully connected
9	9	–	Fully connected
15	15	–	Fully connected
6d	6	1	Top node in diamond hidden from bottom*
6r	6	2	Two top nodes hidden from two bottom nodes* (i.e., only nearest neighbors are connected)
9d	9	1	Opposite points of diamond hidden from each other
9p	9	5	Only nearest neighbors are connected
15d	15	1	Diagonal corners in rectangle hidden from each other
15h	15	6	Upper two rows in rectangle hidden from lower two rows*

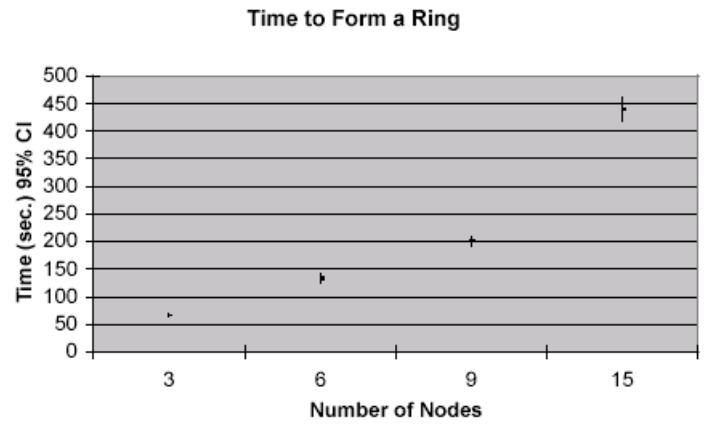
\* and vice versa

#### 4. SIMULATION RESULTS

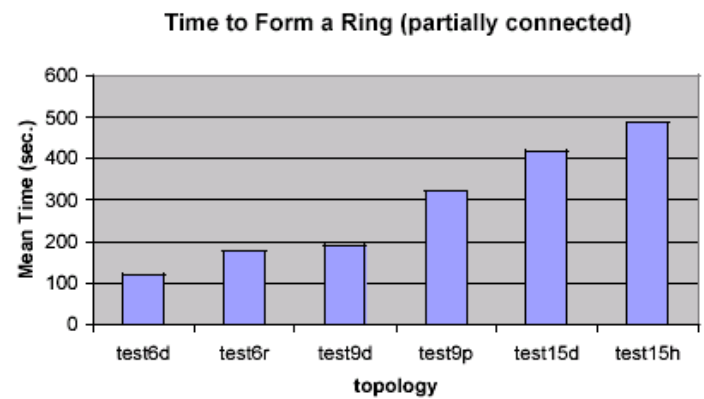
This section presents the simulation results for time to form a ring, starting from all nodes in the reset state. A discussion of these results follows the charts. Each of the topologies in Table 1 was simulated at packet loss rates of 0%, 1%, 3%, and 6%. Space permits us to include only the error-free and 3% results.

In each case, a series of independent simulations was run, and 95% confidence intervals of the time to form a ring in each configuration was computed. In the fully connected cases, the resulting confidence intervals are shown as error bars on the graphs.

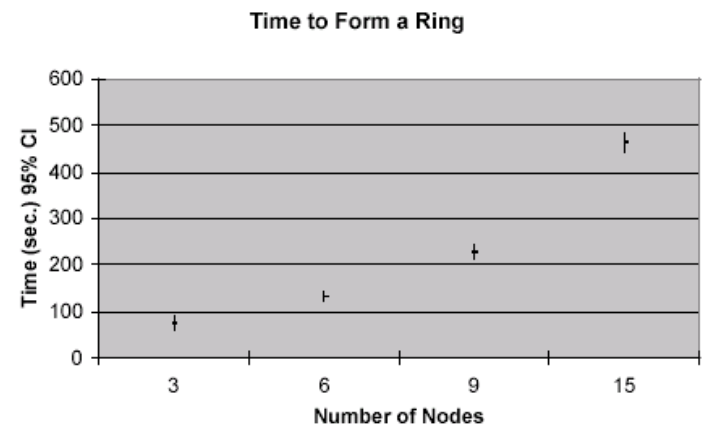
- Figure 5 shows the fully connected topologies in the loss-free channel.
- Figure 6 shows the partially connected topologies in the loss-free channel.
- Figure 7 shows the fully connected topologies with 3% packet loss rate in the channel.
- Figure 8 shows the partially connected topologies with 3% packet loss rate in the channel.



**Figure 5: Fully Connected, Error-Free Case**



**Figure 6: Partially Connected, Error-Free Case**



**Figure 7: Fully Connected, 3% Packet Loss**

Time to Form a Ring (partial connected)

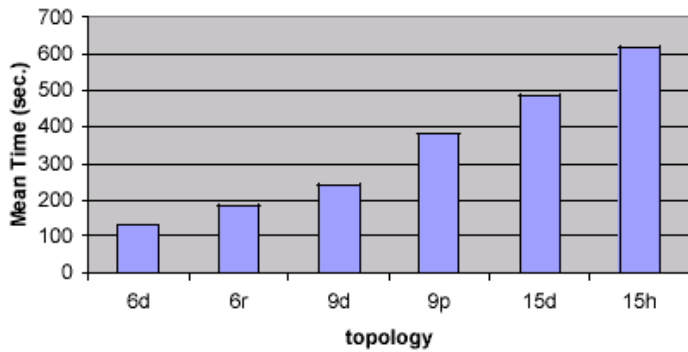


Figure 8: Partially Connected, 3% Packet Loss

In the fully connected topologies, the time to form a ring was essentially linear in the number of nodes for networks of 3 through 9 nodes, independent of the channel packet loss rate. In the error-free case, roughly one minute was required per three nodes in the network. This is roughly twice the token rotation time of the fully formed ring.

The time required to form a ring of 15 nodes was greater than a linear model would predict. This was mainly due to collisions in responses to solicitations, and was expected since the simulations did not increase the number of slots for larger networks (an attempt at realism, since changing this would require manual intervention by shipboard network operators).

The extra time required to form rings in the presence of 3% packet losses amounted to about 10% compared to the error-free case for our fully connected topologies. A 6% packet loss rate resulted in only slightly longer ring formation times.

The partially connected topologies revealed more interesting behavior in the protocol. The presence of hidden nodes allowed formation of multiple rings in the early stages of network startup. Of course, this could not persist, because some nodes could detect both rings and these nodes caused the breakup of one or both rings. As one would expect, the less well-connected topologies were more subject to such false starts and took longer to form rings than better- or fully connected topologies.

However, a curious result emerged in the near-fully connected cases, in which only the nodes at the “corners” of the topology were hidden from each other: these topologies sometimes formed rings *more* quickly than their fully connected counterparts. This appears to be a result of reduced collisions in responses to solicitations.

## 5. HFTP MEASUREMENTS

This section presents laboratory measurements of ring formation time in 3 and 6-node networks. The 3 and 6-node network nodes were brought online with an average sequential startup of 9.5 and 8.4 seconds respectively. A discussion of these results follows the charts. Only the 3 and 6 node topologies in Table 1 were implemented in the lab.

- Figure 9 shows the ring start and join times for each node in the 3 node network
- Figure 10 shows the ring start and join times for each node in the 6 node network

The HFTP was implemented in a laboratory environment by the U.S. Navy SPAWAR Systems Command as a follow on effort to the Battle Force Email STANAG 5066 [5] stack. The HFTP Internet Protocol (IP) Client Stack was designed to implement a full TCP/IP capability and the token bus MAC layer protocol. Due to hardware limitations, the laboratory network configuration is currently limited to 3- and 6-node fully connected topologies. The laboratory configuration of each node includes three computers: an application server and client workstation (running Windows), with the IP Client and MAC Layer protocols running on a Linux platform. Crypto was not used during these measurements, so the associated delays were not present.

The key operating parameters for the protocol in the laboratory implementation were as follows:

Turnaround time	1 second
Token pass time (mean)	9 seconds
Token Holding Time	As high as 3 min
Data rate	9600 bps
Average startup interval	9.5 seconds (3 node) and 8.4 seconds (6 node)
Ring Solicitation	Solicit every 10 rotations, Solicitor rotates
Slots after solicitation	3 (1 second each)
Congestion control	Adaptive response probability

Rather than vary the number of response slots (as in the simulated protocol), this implementation instead uses a fixed number of slots (3), but backs off the probability of responding: after a node fails to connect 3 consecutive times then it will respond to subsequent solicitations only 1/3 of the time.

While the simulation results show the time to form a ring starting from all nodes in the reset state, the laboratory implementation results reveal that the actual ring forming time (RFT) is highly dependent on the node startup time interval. If the nodes are started in a staggered time-wise fashion then collisions will obviously be minimized and the RFT is very predicable. Although the actual node startup intervals are included in the results, a comparative analysis between the actual and simulated start-up interval will not be included In this paper. The RFT graphs presented show the startup-time for each node to ensure that a node is not undeservedly penalized for starting late.

Although controlling the individual node startup sequence is feasible in a laboratory environment, it is most likely that the formation ad hoc network at sea to support a typical Carrier Battle Group (CVBG) may be less controlled. This is due mainly to the fact that ships that compose the CVBG are not initially co-located.

The “first” node to start up in the sequence waits a fixed 10 seconds plus an additional random amount of time (0→ 9 sec) while it listens for a SOLICIT SUCES-SOR (SLS). If it does not hear one then it sends out its own SLS request, goes into the SEEKING (SEK) or listening state, waits 6 seconds, and then enters a SELF RING (SFR) state. All nodes within listening range may reply to the solicit in one of three fixed 1-second time slots. This suggests that if there are one or more nodes listening then the probability of collision is 1/3 and 1/9 for back-to-back

collisions. Nodes that respond to SLS invitation to join the ring do so by sending a SET SUCCESSOR (SET) and enter the JOINING (JON) state after a solicit-reply-timeout. Nodes that lose the contention for the same slot send SetSuccessor at the same time and are forced to wait 26 seconds (TCON timer) and may only SNOOP (monitor) ring activity.

One ring member solicits every 10 token rotations. After a given node solicits, 10 rotations later that node’s Successor solicits. All nodes receive the SLS broadcast and those nodes not already in the ring send a SetSuccessor.

Once invited into the ring, the soliciting node sends a token and waits for an Ack. If no Ack is heard, the soliciting node makes several attempts to resend the token. After the soliciting node receives the ACK token and transitions into the HAVE TOKEN (HVT) state.

If a node fails to connect after three (3) consecutive attempts then the system throttles that node to a fixed probability of join of 1/3: that is, a node will respond to a solicitation with probability 1/3. If the node fails to try to join 3 times then the algorithm is turned off.

Table 2 and Table 3 lists the node start-up sequence followed by the joined sequence in which each node joined the ring (and turn on time) for 3 and 6 node network respectively. Figure 10 and Figure 11 shows the node forming times for the 3 and 6-node fully connected network. The nodes are graphed according to their start-up sequence shown in Table 2 and Table 3.

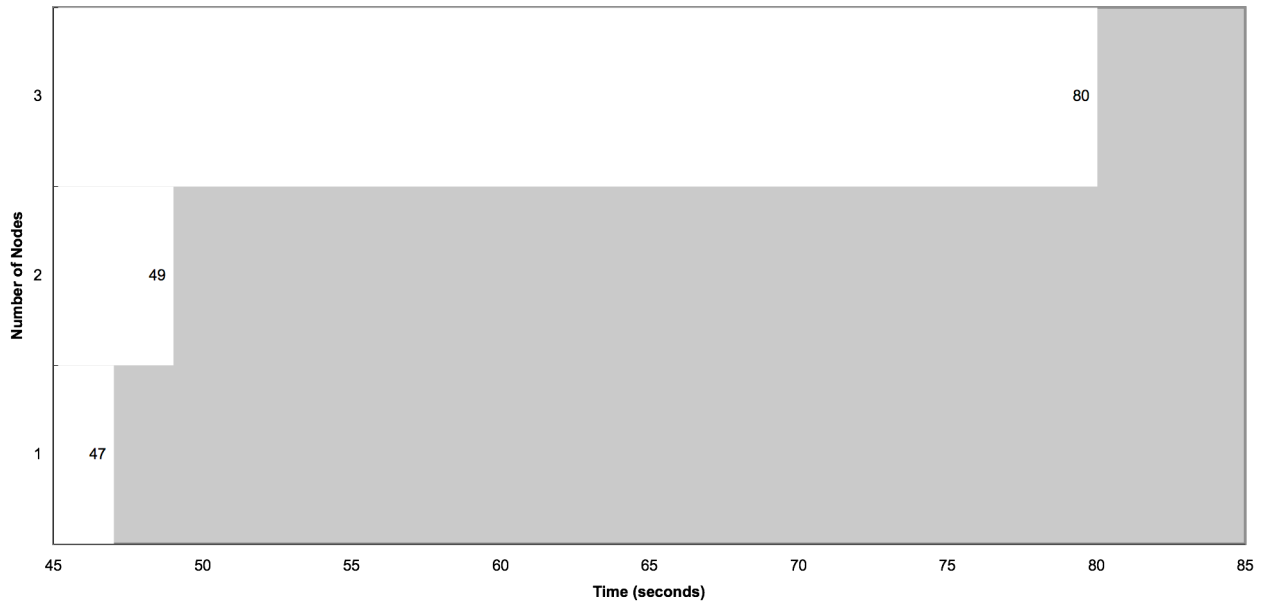
**Table 2: Node Start-up and Join Ring Sequence for 3 Node Network**

Node Start-up Sequence	Node Joined Ring Sequence (Turn on Time)
1	1 (0 sec)
2	3 (19 sec)
3	2 (3 sec)

**Table 3: Node Start-up and Join Ring Sequence for 6 Node Network**

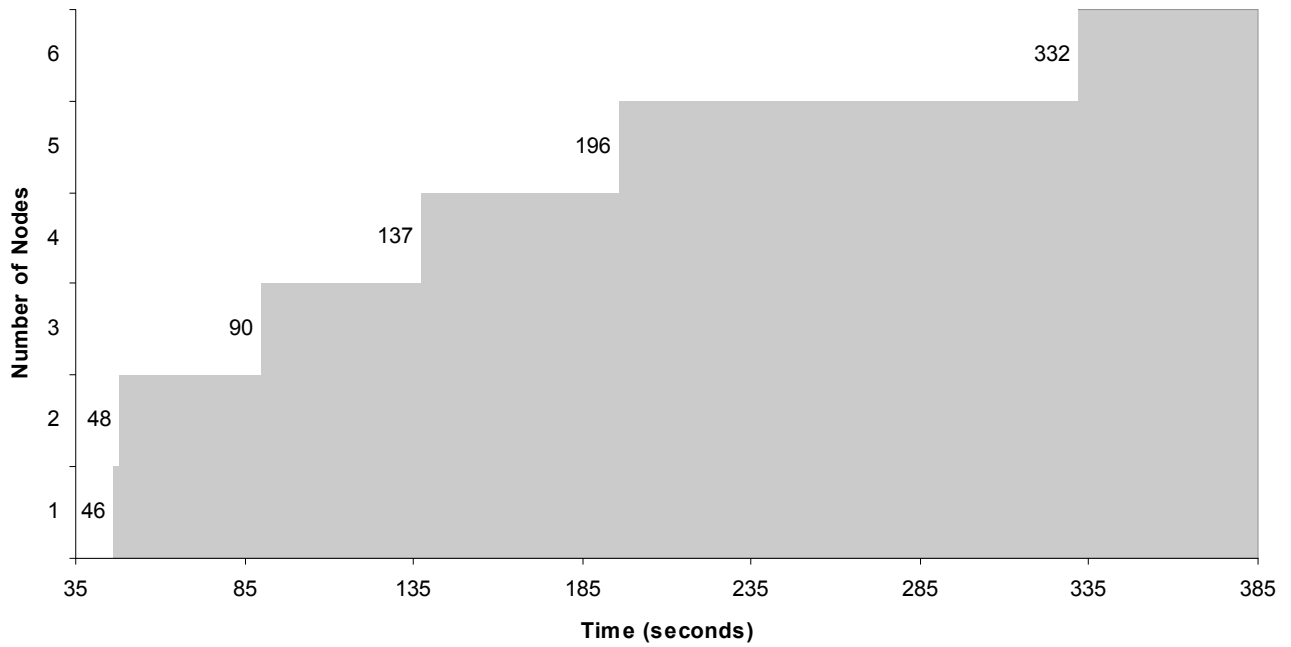
Node Start-up Sequence	Node Joined Ring Sequence (Turn on Time)
1	3 (17 sec)
2	1 (0 sec)
3	4 (4 sec)
4	5 (35 sec)
5	6 (42 sec)
6	2 (9 sec)

**Join Time for 3 Node Network**



**Figure 9: Ring Formation Time (3 Nodes)**

**Join Time for 6 Node Network**



**Figure 10: Ring Formation Time (6 Nodes)**



## 6. CONCLUSIONS

The token passing MAC was able to form rings of up to 9 nodes in startup transients lasting roughly two solicitation rotation times. This held true for packet loss rates from 0 through 6%, and in both fully connected topologies and those with only a few hidden nodes. When only nearest neighbors could communicate, however, ring formation times were extended by about one minute.

The actual ring forming times observed in the laboratory for the 3-node network shows an increase of less than 5% from the simulated value (79 sec actual, 75 mean simulated). This is due mainly to the fact that the 3-node network experienced two slot contentions for node 2 and was restarted, costing the system 28 seconds without which would have resulted in 10% ring forming time decrease. The actual ring forming time for the 6-node network showed a increase of 250% from the simulated value (329 sec actual, 133 mean simulated). The 6-node network appeared to have experienced four slot contentions costing the system approximately 150 seconds without which it would have matched the simulated results. Both networks experienced problems with the same node (node-2) that would not join the ring after all other nodes had already joined possibly indicating a malfunction with that particular node.

## REFERENCES

1. E.E. Johnson, M. Balakrishnan, and Z. Tang, "Impact Of Turnaround Time On Wireless MAC Protocols," *Proceedings of MILCOM 2003*, Boston, MA, 2003.
2. ANSI/IEEE Std 802.11, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 1999.
3. E.E. Johnson, Z. Tang, M. Balakrishnan, J. Rubio, H. Zhang, and S. Sreepuram, "Robust Token Management for Unreliable Networks," *Proceedings of MILCOM 2003*, Boston, MA, 2003.
4. V. Bharghavan, *et al*, "MACAW, A Media-Access Protocol for Wireless LANs," *Proceedings of SIGCOMM '94*, ACM, 1994.
5. M. Balakrishnan and E. Johnson, "Queuing Analysis of DCHF and Token Passing MAC Protocols with Varying Turnaround Times, Proceedings of IPCCC '04, Phoenix, IEEE, April 2004.
6. NATO Standardization Agreement 5066: *Profile for High Frequency (HF) Radio Data Communications*, version 1.2, NATO Standardization Activity reference 0114-C3/5066, 27 January 2004.