A Performance Comparison of the Temporally-Ordered Routing Algorithm and Ideal Link-State Routing

Vincent D. Park
Naval Research Laboratory, USA
vpark@itd.nrl.navy.mil

M. Scott Corson
University of Maryland, USA
corson@isr.umd.edu

Abstract

We present a relative performance comparison of the Temporally-Ordered Routing Algorithm (TORA) with an Ideal Link State (ILS) routing algorithm. The performance metrics evaluated include bandwidth efficiency for both control and data, as well as end-to-end message packet delay and throughput. The routing algorithms are compared in the context of a dynamic, multihop, wireless network employing broadcast transmissions. The network parameters varied include network size, average rate of topological changes and average network connectivity. While the average network connectivity was found not to be a significant factor, the relative performance of TORA and ILS was found to be critically dependent on the network size, and average rate of topological changes. The results further indicate that for a given available bandwidth—as either the size of network increases or the rate of network topological change increases, the performance of TORA eventually exceeds that of ILS.

1. Introduction

The Temporally-Ordered Routing Algorithm (TORA) [1] is a distributed routing algorithm for mobile, multihop, wireless networks which builds upon the earlier work of [2] and [3]. It is best suited for use in large, dynamic, bandwidth-constrained networks such as those proposed for future mobile military systems. TORA is designed to minimize reaction to topological changes. A key concept in its design is that it largely decouples the generation of potentially far-reaching control message propagation from the rate of topological changes. Control messaging is typically localized to a very small set of nodes near the change without having to resort to a dynamic, hierarchical routing solution with its attendant complexity. This localization is achieved at the cost (or benefit?) of not performing a shortest-path routing computation—i.e., TORA does not perform shortest-path routing. However, for the conditions expected in large mobile networks, it will be seen that this approach is superior to link-state routing.

This paper presents a detailed performance comparison of TORA with Ideal Link-State (ILS) routing and pure flooding [4]. Comparison with ILS is useful due to its simplicity and familiarity. Furthermore, ILS technology is the basis for the Open Shortest Path First (OSPF) [5] routing protocol currently under consideration for use by the U.S. military in its large mobile networks. Comparison with flooding is secondary, and is useful to see the network environment in which a more efficient routing technique than flooding is necessary. Comparison with flooding is also useful since one might expect that—as the rate of topological change increases in a dynamic network, eventually all other routing algorithms will essentially “breakdown” leaving flooding as the only recourse. It is useful to know whether our test scenarios are operating at or near this breakdown point.

The paper is organized as follows: a brief description of TORA is given in section 2, a detailed description of the simulation design is given in section 3, the performance results for the scenarios we considered are given in section 4, and some final thoughts and future work are given in section 5.

2. Protocol Overview

Although space constraints prohibit a full description of TORA (see [1] for the full protocol specification), the simulated[1] version can be briefly described as follows. A separate version of TORA runs independently for each destination. The algorithm is distributed in that nodes need only maintain information about adjacent nodes (i.e., one-hop knowledge). It guarantees all routes at any instant in time are loop-free, and typically provides multiple routes for any source/destination pair that requires a route. The protocol is “source initiated” and quickly creates a set

[1] A slight modification to the specification in [1] was necessary to ensure stability between the route creation and route erasure processes in networks with ongoing topological changes. The modification allows only nodes with a “non-reflected” height to generate an UPD packet in response to a QRY reception. Additionally, nodes with “reflected” reference levels must participate in the QRY forwarding process.

of routes to a given destination—only when desired—using a query-reply process which builds a directed acyclic graph of routes rooted at the destination. Since multiple routes are typically established, many topological changes require no reaction at all, as having a single route is sufficient. Following topological changes that do require reaction, the protocol quickly re-establishes valid routes via a temporally-ordered sequence of diffusing computations—each computation consisting of a sequence of directed link reversals. The guiding principle behind the protocol's design is to minimize reaction to topological changes. This principle, in turn, serves to minimize communication overhead. Finally, in the event of a network partition, the protocol detects the partition and erases all invalid routes within a finite time.

Realizing this behavior requires the use of three control packet types: query (QRY), update (UPD) and clear (CLR). QRY packets are used by source nodes to search for nodes that know a route to the intended destination. The search is accomplished via flooding. UPD packets are used to build and maintain routes. During route building and maintenance, each node maintains a value (which can be thought of as a "height") and nodes exchange their heights. Links are assigned a direction based on the heights of neighboring nodes—i.e., they are directed from higher to lower. The significance of the heights is that—during routing, a node may only route information to a lower node. The route building process can be thought of as a directed flood which is generated in response to QRY reception, while the route maintenance process proceeds as a sequence of directed link reversals in response to the loss of some node's last downstream link. CLR packets are used to erase routes. Route erasure occurs when a node perceives[2] that it has detected a network partition. The effect of route erasure at a node is to set its height to null.

3. Simulation Design

A design goal of our simulations was to evaluate the effect of varying the following three network characteristics:

- Network size
- Rate of topological change
- Network connectivity

A well-designed series of tests can provide insight into TORA's applicability for mobile wireless networks. Due to its ability to minimize and localize reactions to topological changes, TORA was expected to outperform ILS as the network size and rate of change were increased. Furthermore, TORA was expected to perform better in densely-connected networks, since this would tend to further minimize and localize its failure reactions.

The relative performance of the three routing methods simulated herein was based on measurement of the following parameters:

- Bandwidth utilization efficiency
  - Number of data bits transmitted per message bit delivered
  - Number of control overhead bits transmitted per message bit delivered
  - Total number of bits transmitted per message bit delivered
- Mean message packet delay
- Message packet throughput (fraction of packets delivered)

These measures were intended to provide insight into the ability of the protocols to route packets to their intended destination, and the efficiency of the protocols in accomplishing that task.

The comparison of TORA, ILS and pure flooding was accomplished via simulation using the Optimized Network Engineering Tool (OPNET). In order to provide sufficient control of the networking environmental characteristics, and to permit simulation of the largest possible networks in a reasonable time using OPNET, a mobile wireless network was modeled in the simulations using a *fixed* network topology with the ability to control failure/recovery of individual links. Links in the fixed topology essentially indicate radio connectivity between node pairs. In the model, failures/recovery of each link was determined randomly and independently. Multiple base topologies were used to vary the size of network, while simulation parameters (input at runtime) were used to adjust the rate of topological change, the network connectivity, and the message traffic load. Each of the three routing protocols was implemented over a common framework, and every effort was made to ensure that any simplifying assumptions did not favor one protocol over another.

The common framework consisted of the network model and the node model, which were constructed on top of an assumed data link layer. The details of the network topological design, link failure/recovery mechanics, link error/delay characteristics, and channel access are discussed in the network model section. The node model section covers the details of the message traffic generation, queueing mechanics, and transmission delays.

[2] A node's perception of a partition is not always correct. Routes that are no longer valid (i.e., rooted at the destination) may sometimes be erased in the absence of an actual physical partition. A node may perceive a partition when it is no longer connected to the destination via a path of assigned (i.e., directed) links—although it may still connected to the destination via some path of unassigned (i.e., undirected) links.

3.1. Network Model

Three different baseline network topologies (19, 43 and 99 nodes) were used to vary network size. In order to avoid any bias in the results due to changes in "network connectivity" as a function of network size, all of the networks were designed with a similar topological structure—i.e., a densely-connected honeycomb shown in Figure 1(a).

Each link in a given network continuously cycled between two states (ACTIVE and INACTIVE) independently of all other links. Once ACTIVE, the time a link remained ACTIVE was determined randomly based on an exponential distribution. The mean of the distribution ("mean-time-to-failure," $1/\mu$) was an input parameter of the simulation. Running successive simulations, while varying this parameter, allowed evaluation of the effect of the topological change rate on routing performance. Essentially, a lower link mean-time-to-failure corresponded to a higher rate of topological change. The long-term average fraction of time each link would remain operational, f, was also a simulation input parameter. Variation of this parameter affected the average overall network connectivity (i.e., when $f = 0.5$, on average 50% of the links in the network are operational at any given time). A "snapshot" of how the 43-node topology may look at an instant in time is depicted in Figure 1(b). The parameter f was also used to determine the initial state of each link at the beginning of each simulation execution. Once INACTIVE, the time a link remained INACTIVE was also determined randomly by an exponential distribution. However, the mean of the distribution ("mean-time-to-repair," $1/\lambda$), was computed from $1/\mu$ and f. The state transition diagram for this continuous-time Markov process, and the equation by which $1/\lambda$ is computed, are presented in Figure 2.

Each ACTIVE link permitted error-free transmission in either direction, and we assumed that channel access is handled at the link level. The link propagation delay was set to zero, since it is relatively insignificant when compared to the packet transmission and queueing delays When a node needed to "broadcast" a packet to its neighbors, copies of the packet were forwarded over each of its ACTIVE adjacent links. For accounting purposes, when computing the number of bits transmitted, each "broadcast" was counted only once—even though in the simulation, a separate copy had to be delivered to each neighbor. Since channel access was not directly implemented in the simulation, it was possible for a node to receive multiple packets (from different neighbors) simultaneously.

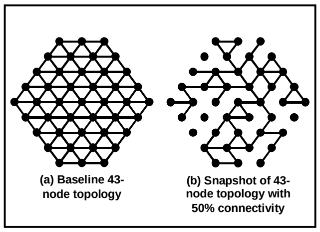
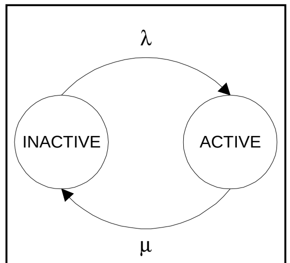


Figure 1. Example baseline topology and snapshot of topology with 50% connectivity



P[ACTIVE] = f

P[INACTIVE] = 1 - f

$$\frac{1}{\lambda} = \frac{(1-f)}{f} * \frac{1}{\mu}$$

Figure 2. Link, state transition diagram

3.2. Node Model

A common node model was used for all of the network nodes. Message packets with a payload size of 1024 bits were produced randomly with exponentially distributed interarrival times by a packet generator in each given node. The mean interarrival time for message packet generation was set to a common value for all of the nodes in a given baseline network topology. The method used to select the mean interarrival times will be described in section 4. Once a message packet was generated, its destination was selected randomly from the set of other nodes in the network (using a uniform distribution) and the packet was queued for transmission. Thus, each node randomly generated message traffic for all other nodes. Control packets were given strict priority over message packets, and message packets bound for differing destinations were serviced by a round-robin policy.

For the ILS implementation the "next-hop" neighbor was selected based on Dijkstra's shortest path computation [4]. While the ILS implementation computed only a single next-hop neighbor for a given destination, TORA often provides multiple "downstream" neighbors for routing to a given destination. Therefore, two slightly different versions of TORA were implemented. In the first version (labeled TORA), the next-hop neighbor was selected randomly from the set of downstream neighbors using a uniform distribution. In the second version (labeled TORA LN) the "lowest" downstream neighbor (as determined by the height of the neighbors) was selected as the next-hop neighbor.

Provided that there was an ACTIVE link over which to transmit, packets were transmitted consecutively without intermediate processing delays. Any packet being transmitted over a link when a failure occurred was considered lost and was discarded by the receiving node. Transmission delay was determined by the length of the packet and a fixed transmission rate that was set to 1024 bits per second. This rate was selected for two reasons. First, for the large, mobile military networks that motivated development of this protocol, raw transmission rates are very low (approx. 10 kbps). After accounting for the overhead associated with channel access, data link control, error detection/correction, etc., usable bandwidth in these networks is on the order of one kbps. Second, simulation of higher transmission rates, while desirable for future studies regarding the effect of bandwidth availability, was not necessary for the relative performance evaluation carried out here. Evaluation of routing performance under higher traffic loads would have required processing of many more simulation events. The additional processing burden would have placed an unreasonably low limit on the network size and rate of change that could be evaluated, due to excessively long simulation run times. Thus, in summary, end-to-end message packet delay was solely a function of route availability, route selection, queueing delays and transmission delays.

4. Results

Testing was completed by executing several sequences of simulations on each given baseline network topology. In each sequence, one input parameter (e.g., link mean-time-to-failure or average network connectivity) was varied while the other parameters were kept constant. For each set of input parameters, all of the routing protocols were subjected to an identical sequence of random events. For each baseline network topology (e.g., 19, 43 and 99-node networks), a suitable message traffic load was selected as follows. A common, mean interarrival rate for message packet generation in all nodes was selected just *above* the threshold where flooding message packets caused significant queueing delay. This corresponded to an environment where a more efficient routing algorithm than pure flooding could provide a lower, average packet delay. The traffic loads for the three networks were as follows: 4.0 packets/node/minute for the 19-node network, 1.5 packets/node/minute for the 43-node network, and 0.6 packets/node/minute for the 99-node network.

4.1. Network Size and Rate of Topological Change

Running a similar sequence of simulations (over varying rates of topological change) on each of the different baseline network topologies, provided insight into the effects of network size and rate of topological change on routing performance.

4.1.1. Network size: 19 nodes. The first sequence of simulations was run on the 19-node network over varying rates of topological change, while the average network connectivity was held constant at 90%. The link mean-time-to-failure was initially set to 32 minutes, and reduced by one half for each successive simulation to one minute. Recall that a smaller link mean-time-to-failure corresponds to a higher rate of topological change. Each data point collected was based on two hours of simulated operation time. The bandwidth utilization as a function of rate of topological change is depicted in Figure 3. The solid (lower) portion of the stacked bars represents the average number of data bits transmitted per data bit delivered (DATA), while the hashed (upper) portion represents the average number of control overhead bits transmitted per data bit delivered (CTRL). The DATA portion represents only the message packet, payload bits, while the CTRL portion represents the control packet bits as well as message packet overhead (header) bits. The solid portion can also be interpreted as the average number of times a message packet was transmitted, or as the average path length (in hops) traveled by data packets (for ILS and TORA).

The results clearly indicate that the average path length traveled by ILS message packets is shorter. However, it is apparent that as the rate of change increases, the amount of control overhead for ILS increases much more rapidly than for TORA. In fact, at the higher rates of change depicted, ILS utilizes more bandwidth for control overhead than for data. The effect that this increased control overhead has on the mean message packet delay is depicted in Figure 4. The mean packet delay has been plotted on a logarithmic scale to provide visual separation between the data points when the delay was less than 10 seconds.

As the rate of topological change increases, the increase in ILS control overhead begins to cause an increase in the average, message packet delay. When the link mean-time-to-failure is less than approximately eight minutes, the average, message packet delay for TORA LN is less than ILS. Thus, despite the shorter average path length traveled by ILS message packets, the additional queueing delay (caused by the excessive ILS control overhead competing “in-band” for the same channel) results in a longer overall message packet delay.

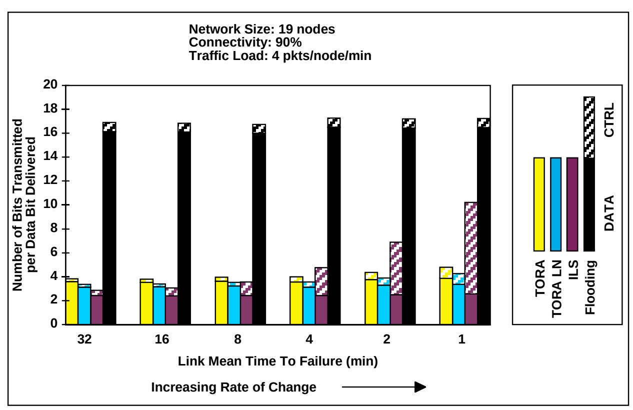


Figure 3. Bandwidth utilization as a function of rate of topological change—19 nodes with 90% connectivity

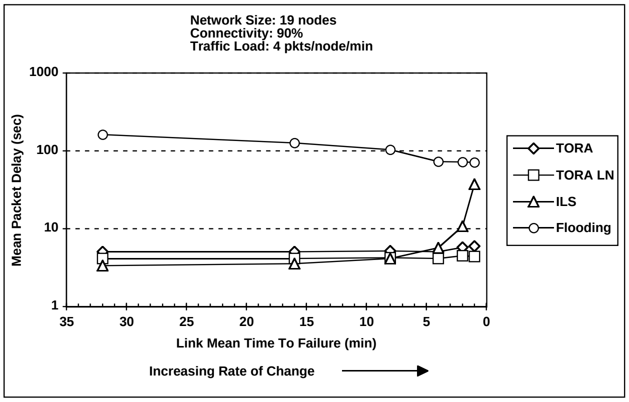


Figure 4. Mean message packet delay as a function of rate of topological change—19 nodes with 90% connectivity

4.1.2. Network size: 43 nodes. The second sequence of simulations was run on the 43-node network over varying rates of topological change, while the average network connectivity was held constant at 90%.

With the exception of some minor differences, the plots for the 43-node network appear very similar to the equivalent plots for the 19-node network. Again, it is clear that the average path length traveled by ILS message packets is shorter. As expected, the average path length traveled by message packets and the minimum mean message packet delay are slightly greater for all of the algorithms in the 43-node network than in the 19-node network. Once again, as the rate of topological change increases, the ILS control overhead increases more rapidly (Figure 5), causing an increase in the average, message packet delay (Figure 6).

Perhaps the most significant difference between the plots for the 43-node and 19-node networks is the rate at which these behaviors are exhibited. For the 43-node network, the average message packet delay for TORA LN is less than ILS when the link mean-time-to-failure is less than approximately 150 min. (2 hr. 30 min.). This begins to show the effect of network size.

Since ILS must maintain full topological knowledge at all nodes, the scope of the failure reactions (and thus the amount of control overhead associated with each topological change) increases with network size. Furthermore there is a coupling between the aggregate rate of topological change in the network and network size. Since the larger network has a greater number of links, if the link failure/recovery mechanics are constant (i.e., same average connectivity and link mean-time-to-failure), the larger network will experience a greater aggregate rate of network topological change. Because of these factors, ILS cannot tolerate as high of a rate of change in the larger network. Since TORA tends to localize its reactions to topological change, it is not as significantly affected by the increase in network size.

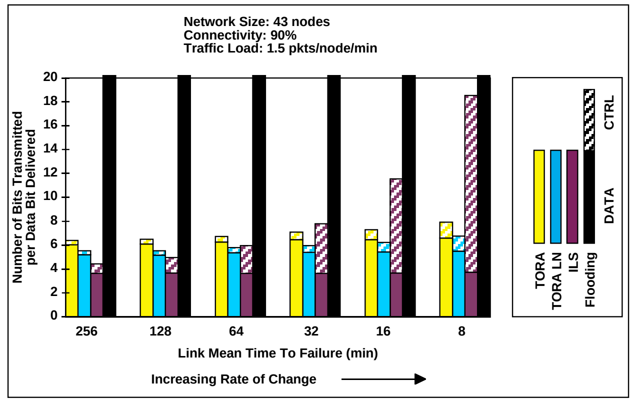


Figure 5. Bandwidth utilization as a function of rate of topological change—43 nodes with 90% connectivity

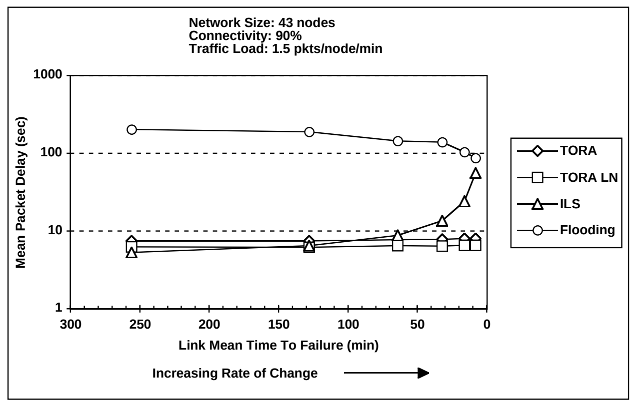


Figure 6. Mean message packet delay as a function of rate of topological change—43 nodes with 90% connectivity

4.1.3. Network size: 99 nodes. The third sequence of simulations was run on the 99-node network over varying rates of topological change, while the average network connectivity was held constant at 90%.

The two plots (Figures 7 and 8) illustrate the expected results. The rate threshold (where TORA LN begins to outperform ILS) occurs at even a lower rate. In fact, TORA LN continues to provide a lower mean message packet delay even when the link mean-time-to-failure is approximately 273 hours. This is potentially due, in part, to the large amount of control overhead required for initialization when using ILS.

The mean message packet delay for TORA and TORA LN appears to be increasing slightly at the highest rates of topological change depicted. However, a close evaluation of Figure 7 and the corresponding plots for the 19-node (Figure 3) and 43-node (Figure 5) networks suggests that this may not be due to an increase in TORA control overhead. The ratio of CTRL to DATA bits for TORA did not increase more for the 99-node network than for the smaller networks. However, the average path length traveled by TORA message packets in the 99-node network shows a slightly disproportionate increase. Thus, the slight increase in average path length is probably the cause of the slight increase in delay.

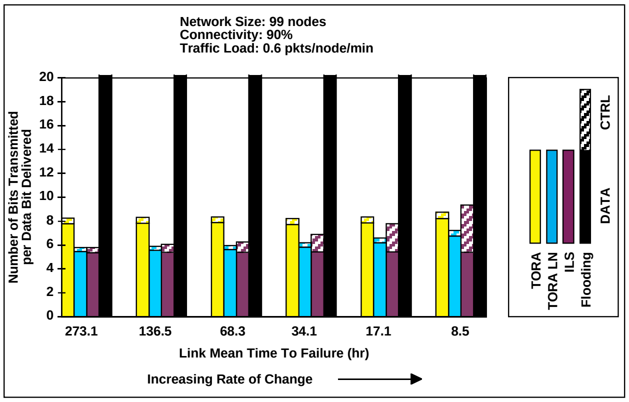


Figure 7. Bandwidth utilization as a function of rate of topological change—99 nodes with 90% connectivity

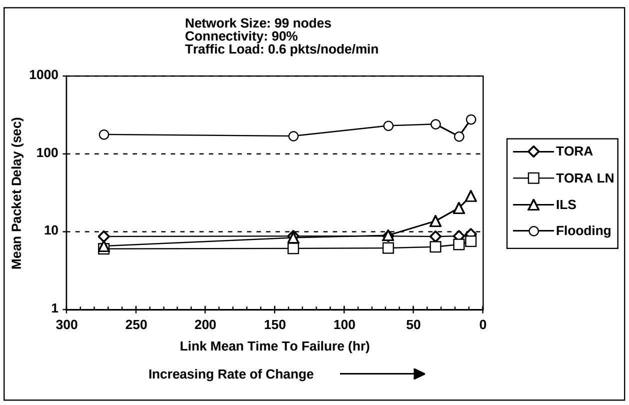


Figure 8. Mean message packet delay as a function of rate of topological change—99 nodes with 90% connectivity

4.2. Network Connectivity

A sequence of simulations was run over a range of average network connectivity, while the link mean-time-to-failure was held constant. The conditions of this starting point were an average network connectivity of 90% and link mean-time-to-failure of 32 minutes. In each subsequent simulation the average connectivity was reduced by 10%. The bandwidth utilization and mean message packet delay as a function of average network connectivity are depicted in Figures 9 and 10.

Note that while the ratio of CTRL to DATA bits for TORA increases slightly, the ratio for ILS decreases. The decrease in ILS control overhead results from a coupling between the *aggregate* rate of network topological change and average network connectivity. The average number of link-state changes per minute for the 90% and 40% connectivity simulations (measured during simulation execution) are depicted on the plots (5.9 changes/min. and 2.7 changes/min. respectively).

Assuming the amount of control overhead was the only factor affecting mean message packet delay, one might expect the delay for TORA packets to increase and the delay for ILS packets to decrease (with decreasing average network connectivity). However, Figure 10 illustrates that this is not the case. As the average network connectivity decreases, the mean message packet delay increases dramatically for both TORA and ILS. The mean delay for flooding packets, on the other hand, decreases.

The likely cause for this result is an increased number of network partitions. In a more sparsely-connected network, a given node is more likely to be unreachable by some other set of nodes for a longer period of time. Since the TORA and ILS protocol implementations essentially queue message packets until a route becomes available, this can have a significant effect on the mean message packet delay. Alternatively, the scope of flooding message packets would tend to be smaller, on average. This reduces mean packet delay for flooding in two ways. First, the average number of packets to be processed by a given node (over time) would be less, resulting in less queueing delay. Second, packets with the least number of hops required to reach their intended destination would be most likely to be delivered.

Although throughput plots have not been included herein, the conjecture of "an increased number of network partitions" is further supported by the throughput measurement data. Despite the lower aggregate rate of network topological change associated with the more sparsely-connected networks, the throughput for flooding packets decreases dramatically.

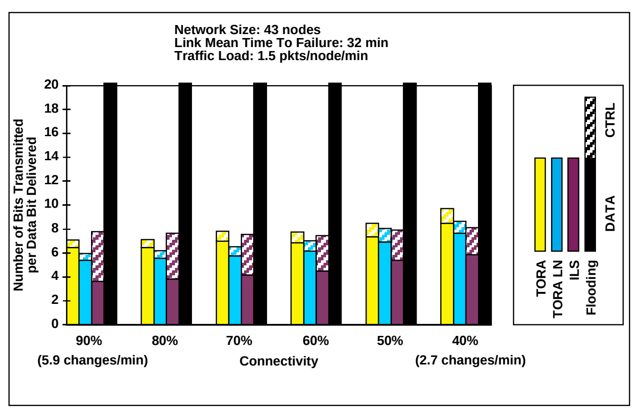


Figure 9. Bandwidth utilization as a function of average network connectivity—43 nodes with 32 min. link mean-time-to-failure

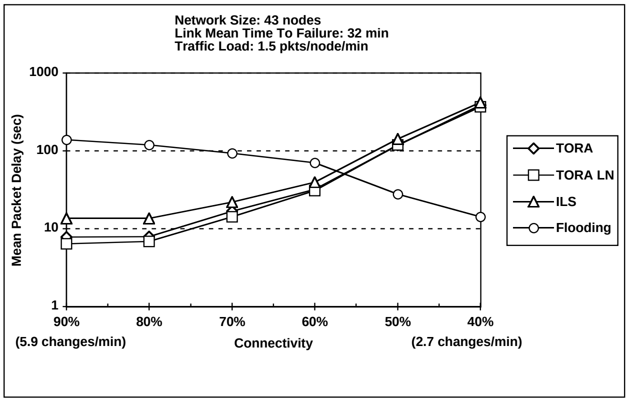


Figure 10. Mean message packet delay as a function of average network connectivity—43 nodes with 32 min. link mean-time-to-failure

5. Conclusions

An simulation study was conducted to evaluate the relative performance of TORA and ILS routing. The simulations were designed to provide insight into the effect of varying network size, average rate of topological changes and average network connectivity. While the average network connectivity was found not to be a significant factor, the relative performance of TORA and ILS was found to be critically dependent on the network size, and average rate of topological changes.

The results indicate that for a given available bandwidth—as either the size of network increases or the rate of network topological change increases, the performance of TORA eventually exceeds that of ILS. Specifically, as the network size and/or rate of topological change increases, the amount of control overhead for ILS increases much more rapidly than for TORA—effectively, congesting the communication channel and causing additional queueing delay for message traffic. Therefore, above some combination threshold of network size and rate of topological change, TORA provides lower end-to-end message packet delay on average.

The point to be emphasized is that under some networking conditions TORA—which is *not* a shortest-path routing algorithm—can outperform a shortest-path routing algorithm. ILS is but one approach for performing shortest-path routing. There are other approaches—most notably, distributed distance-vector [4, 6, 7, 8, and 9] and path-finding [10 and 11] algorithms. Nevertheless, for a given network size and rate of topological change, any shortest-path algorithm requires a minimum amount of control overhead to permit computation of the shortest-path. We conjecture that as the network size and/or rate of topological change are increased, this minimum amount of control overhead to permit computation of the shortest-path will increase more rapidly than the amount of control overhead for TORA. If so, then there must be some threshold for network size and/or rate of topological change in which any shortest-path routing protocol would perform poorly relative to TORA. This conjecture is difficult to prove in general, but can perhaps be shown to be valid for other existing shortest-path protocols. Future work will compare TORA against these other protocols in an attempt to further support this conjecture.

References

[1] V. Park and M. S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks", *Proc. IEEE INFOCOM '97,* Kobe, Japan (1997).

[2] E. Gafni and D. Bertsekas, "Distributed Algorithms for Generating Loop-free Routes in Networks with Frequently Changing Topology," *IEEE Trans. Commun.* (January 1981).

[3] M. S. Corson and A. Ephremides, "A Distributed Routing Algorithm for Mobile Wireless Networks," *Wireless Networks* 1 (1995).

[4] D. Bertsekas and R. Gallager, *Data Networks* (Prentice-Hall, 1987).

[5] J. Moy, "OSPF Version 2," Internet RFC 1583, Proteon, Inc., March 1994.

[6] P. Merlin and A. Segall, "A Failsafe Distributed Routing Protocol," *IEEE Trans. Commun.* (September 1979).

[7] J. Jaffe and F. Moss, "A Responsive Distributed Routing Algorithm for Computer Networks," *IEEE Trans. Commun.* (July 1982).

[8] J. J. Garcia-Luna-Aceves, "Loop-free Routing Using Diffusing Computations," *IEEE Trans. Networking* 1(1) (1993).

[9] C. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance Vector Routing (DSDV) for Mobile Computers," *ACM SIGCOMM*, (October 1994).

[10] P. Humblet, "Another Adaptive Shortest-Path Algorithm," *IEEE Trans. Commun.* (June 1991).

[11] S. Murthy and J.J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks," *ACM Mobile Networks and Applications Journal*, Special issue on Routing in Mobile Communication Networks, (1996).