Enhancing the TORA Protocol using Network Localization and Selective Node Participation

Kwan Hui Lim and Amitava Datta
School of Computer Science and Software Engineering
The University of Western Australia, Crawley, WA 6009, Australia
Email: kwanhui@graduate.uwa.edu.au, datta@csse.uwa.edu.au

Abstract—**The Temporally-Ordered Routing Algorithm (TORA) is a distributed routing protocol that is based on a family of link reversal algorithms. TORA is able to provide multiple loop-free routes to any destination using the route creation, maintenance and erasure functions. TORA performs well in networks with a small number of traffic connections but poorly in networks with a large number of traffic connections. This poor performance is due to the traffic congestion caused by excessive route maintenance. This traffic congestion is further aggravated by routing overhead produced by the large number of traffic connections. We propose two modifications to improve TORA using a network localization approach and selective node participation approach. The network localization approach initializes and maintains a localized portion of the entire network while the selective node participation approach selects a subset of nodes to participate as part of the network. Benchmarks against original TORA show that our TORA modifications results in an overall performance improvement in terms of packet delivery, routing overhead and packet latency.**

I. Introduction

TORA is a highly adaptive distributed routing algorithm that is designed to work in Mobile Ad hoc Networks (MANET) where the network topology is dynamic due to the presence of mobile nodes. TORA is also able to provide multiple loop-free routes to any destination on-demand using three functions of route creation, maintenance and erasure. TORA is layered on top of the Internet MANET Encapsulation Protocol (IMEP) and uses the link/connection status sensing, broadcast reliability, and message aggregation services provided by IMEP.

Broch et al. and Broustis et al. [1], [2] showed that TORA performs poorly when there is a large number of traffic connections. They observe that this poor performance is due to the route maintenance process where temporary loops may be formed thus causing data packets to be dropped. Other authors made similar observations about the presence of temporary loops in TORA [3]–[5]. Specifically, Broch et al. [1] and Broustis et al. [2] observed how TORA and IMEP contribute to network congestion that causes the poor performance of TORA. With a congested network, IMEP incorrectly detects that links are broken and informs TORA, which initiates route maintenance. Consequently, route maintenance produces even more overhead thus congesting the network and causing IMEP to incorrectly detect more links as broken.

Lim and Datta [6] studied the effects of IMEP on the TORA protocol and suggested modifications to IMEP to improve the performance of TORA. These IMEP modifications include introducing a random retransmission period and extending the retransmission period for the object block messages. Despite improvements at the IMEP level, modifications to TORA itself could result in greater improvement. At the TORA layer, Dharmaraju [7] proposed querying for routes using an incremental ring search in order to reduce routing overhead. Instead of flooding the entire network with route queries, they do so with an increasing number of hops until a route is found. However, this approach requires multiple repetitions of route querying for every single destination, even more so if the destination node is far away. Similarly, Coll-Perales and Gozalvez [8] proposed a method to localize routes to the shortest path between a source and destination. However, this proposal is based on the Ad-Hoc On-Demand Distance Vector protocol, whose operations are distinctly different from TORA.

Our main contributions include proposing two modifications to TORA: the network localization approach; and selective node participation approach. The network localization approach initializes and maintains only a subset of the network, and the selective node participation approach chooses a subset of nodes to participate in route creation, maintenance and erasure. These approaches differ from that of Dharmaraju in that we require only a single pass of route query instead of multiple route queries in an expanding ring search manner. We also show that our modifications improve the overall performance of TORA in terms of packet delivery, routing overhead and average packet latency.

II. Temporally-Ordered Routing Algorithm

TORA is a distributed routing algorithm that is based on a family of link reversal algorithms and is able to provide multiple loop-free routes to any destination on-demand. The availability of multiple paths is a result of how TORA models the entire network as a directed acyclic graph (DAG) rooted at the destination. Each node has a height associated with it and links between nodes flow from one with a higher height to one with a lower height. The collection of links formed between nodes forms the DAG and ultimately all nodes will have a route to the destination. For each possible destination required, a separate DAG needs to be constructed.

TORA works on the premises that links between nodes are bi-directional, nodes are always aware of their neighbouring nodes, packets are received reliably in the correct order, and that broadcasting is used. TORA depends on IMEP [9] to ensure that these premises are valid, by using the link/connection status sensing, broadcast reliability, and message aggregation services provided by IMEP. The following description of TORA is taken from Park and Corson [5].

A. *Route Creation*

The three main functions carried out by TORA are route creation, maintenance and erasure. Initially, all nodes start off with a null height and links between the nodes are unassigned. When a node requires a route to a destination, it initiates route creation where *query* packets are flooded out to search for possible routes to the destination. Eventually, a *query* packet reaches either a node that has a route or the destination itself, and the node replies with an *update* packet. When a node receives an *update* packet, it sets its link as directed from itself to the sender of the *update* packet. This setting of directional links eventually reaches the node which requires the route and provides it with at least a route to the destination.

B. *Route Maintenance*

Route maintenance occurs when a node loses all of its out-going links (thus all of its possible routes to a destination). This can be caused by either the detection of a link failure or link reversal following the receipt of an *update* packet. When the detection of a link failure causes a node to lose all of its out-going links, the node propagates an *update* packet which reverses the links to all of its neighbouring nodes. Intermediate nodes that receive the *update* packet then reverse the links of their neighbouring nodes. Links are reversed only for neighbouring nodes that do not have any out-going links and have not performed link reversal recently. The link reversal needs to be repeated until each node has at least one out-going link. This entire process ensures that the DAG is maintained such that all nodes have routes to the destination.

The route maintenance function of TORA is the main problem as this function produces a large amount of routing overhead. This large amount of routing overhead causes the network to be congested thus preventing data packets from reaching their destinations. As mentioned earlier, route maintenance is initiated upon the discovery of a link failure that causes a node to lose all of its out-going links. Our modifications aim to restrict route maintenance to only the segment of the network that actively participates in data communication, thus reducing routing overhead and allowing for the delivery of data packets.

C. *Route Erasure*

In the event that a node is in a network partition without a route to the destination, route erasure is initiated. The detection of a network partition is undertaken by the node that first initiated route maintenance (i.e. due to a link failure causing the loss of all its out-going links). During route maintenance, the node sends out *update* packets to reverse links to all its neighbouring nodes and attempts to find a route to the destination. It is able to determine the presence of a network partition if a similar *update* packet is sent back to it by another node. This means that all nodes in the current network partition cannot find a route and are trying to find a route through the original node. Route erasure is then performed by the node by flooding *clear* packets throughout the network. When a node receives a *clear* packet, it sets the links to its neighbours as unassigned. Eventually, these *clear* packets propagate through the network and erase all routes to that unreachable destination.

III. Experimental Setup

We evaluated our proposed TORA modifications with original TORA using the *ns-2* network simulator [10]. *Ns-2* is a discrete event simulator that allows for the modelling of a variety of protocols over wired, wireless and satellite networks. This simulator allows us to compare the performance of our proposed modifications with that of original TORA in terms of packet delivery, routing overhead and average packet latency. We modelled the radio propagation properties of mobile nodes using the Lucent WaveLan direct sequence spread spectrum radio [11]. We also implemented the IMEP modification of extending the retransmission time of the object block message and adopted an experimental setup similar to that in [6].

The movement scenarios for nodes in the network are modelled using the random waypoint model [12]. In this model, nodes start off at a random position before moving to a random destination at a random speed between 0 and some pre-defined maximum. We define this maximum as 1 m/s or 20 m/s to model a low and high mobility network respectively. Upon reaching the destination, it stops for a certain *pausetime* before moving to another random destination. This takes place for the entire duration of the simulation, set at 900 seconds.

The topology size of the network was chosen at 1,500m × 300m for a network of 50 nodes, and 2,121m × 425m for a network of 100 nodes to maintain a node density of approximately 9,000m^2 per node. We designated the pause times as 0, 30, 60, 120, 300, 600, and 900 seconds to model different levels of mobility from constant to no movement.

Constant bit rate (CBR) sources are used as traffic sources where they constantly send packets to a particular destination. In our case, CBR packets of 64 bytes are constantly sent at a rate of 4 packets per second, until the end of the simulation.

For each combination of pausetime, mobility and number of traffic connections, the results presented are based on the average of 10 and 20 simulation runs respectively for the 100 and 50 nodes network. Using the results averaged from multiple simulation runs helps to overcome the sensitivity of the simulation results on the movement scenario used.

IV. Heuristics to Improve TORA Protocol

Even with the improvements obtained with a more reliable IMEP [6], there are still problems with the TORA protocol itself and we suggest solutions to fix it in this paper. The process of route creation involves the propagation of QRY packets that initializes the entire network into a DAG but communication rarely involves nodes at the extreme ends of

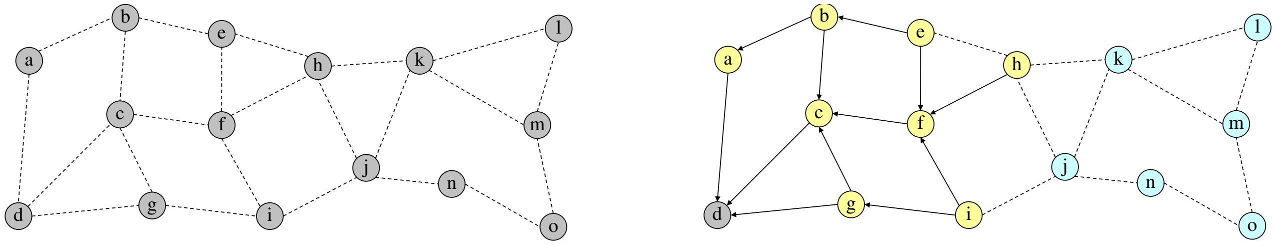

Fig. 1. The initially uninitialized network (left) where node *e* needs a route to node *d*. After route creation using the network localization approach, we obtain a localized network (right) formed by the yellow nodes. The blue nodes are too far away and thus were not initialized.

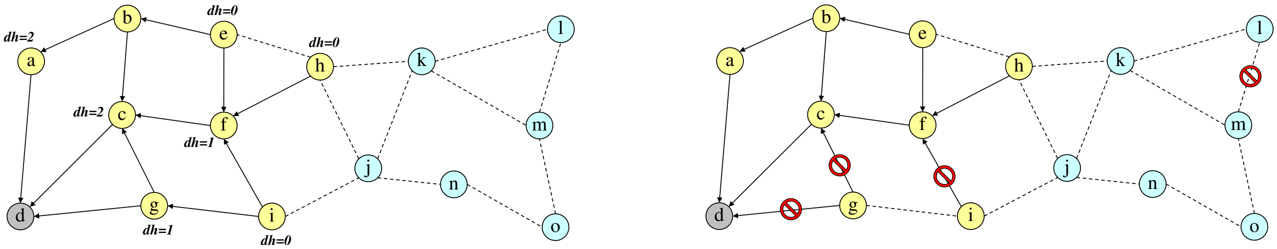

Fig. 2. The localized network with *DAG_hopcount* labelled (left). The breaking of links (*g*, *d*), (*g*, *c*) and (*i*, *f*) causes route maintenance only in the localized network (i.e. setting of link (*i*, *g*) as undirected). This route maintenance does not propagate beyond the localized network due to the *DAG_hopcount* set during route creation. Similarly, link failures outside of the localized network (i.e. link (*l*, *m*)) does not affect nodes in the localized network.

a DAG. A good example is when a node (the source) needs to communicate with another nearby node (the destination) but the DAG extends to nodes that are far away and do not participate in the communication between the source and destination. In the DAG created by original TORA, route maintenance can be triggered by nodes that do not participate in communication. This makes maintaining the DAG an expensive operation due to the unnecessary routing overhead produced. Our proposed modifications are generally based on the idea of reducing the size of the DAG that needs to be maintained. A smaller DAG results in less routing overhead thus allowing more data packets to be sent to their destination.

A. Network Localization Approach

This modification works on the concept of initializing and maintaining a DAG that represents a portion of the network termed the localized network. Initializing a portion of the network reduces the routing overhead that could be caused by distant nodes that do not participate in communication. This approach implements the following features:

- A *propagated_hopcount* (*ph*) variable in QRY packets that keeps track of the number of hopcounts travelled by the packet.
- A *DAG_hopcount* (*dh*) variable in each node that represents how far the localized network stretches and also indicates how far UPD packets should be propagated. A unique *DAG_hopcount* is associated with each DAG.
- An *expiry_hopcount* (*eh*) variable in UPD packets that determines the number of hopcounts a packet will be propagated.

A node *i* that requires a route to a destination broadcasts a QRY packet with *propagated_hopcount* set to 0. This *propagated_hopcount* is incremented each time a node propagates the QRY packet. When a node with a valid route receives the QRY packet, it records *DAG_hopcount* as the value of *propagated_hopcount* in the QRY packet and propagates an UPD packet with *expiry_hopcount* set to *DAG_hopcount*. When a node receives the UPD packet, it records *DAG_hopcount* as the value of *expiry_hopcount* in the UPD packet, decrements *expiry_hopcount* and propagates the UPD packet if *expiry_hopcount* is greater than 0. This modification prevents the UPD packet from propagating too far and initializing the entire network.

Fig. 1 shows an example of how the network is initialized into a localized portion during route creation. Fig. 2 illustrates the process of route maintenance in the event of a link failure and how this process is restricted to only a localized portion of the network. There is no change to the route erasure function.

B. Selective Node Participation Approach

This modification chooses a subset of nodes from the entire network to participate in route creation, maintenance and erasure. Networks with a moderate to high node density are best suited for this modification, where a node *i* has multiple neighbours since many nodes will be within the transmission range of node *i*. After route creation, the failure of node *i* possibly results in route maintenance for all neighbours of node *i* thus causing a substantial amount of overhead in terms of UPD packets. This modification aims to reduce the number of nodes that are redundant to the DAG but still maintain the overall integrity of the DAG (i.e. the existence of multiple routes to the destination). This approach implements the following features:

- A *node_status* variable stored in nodes (with respect to a particular destination) that states whether it participates as part of the network.
- A *probability_active* constant that determines the probability that a node is assigned an active status.

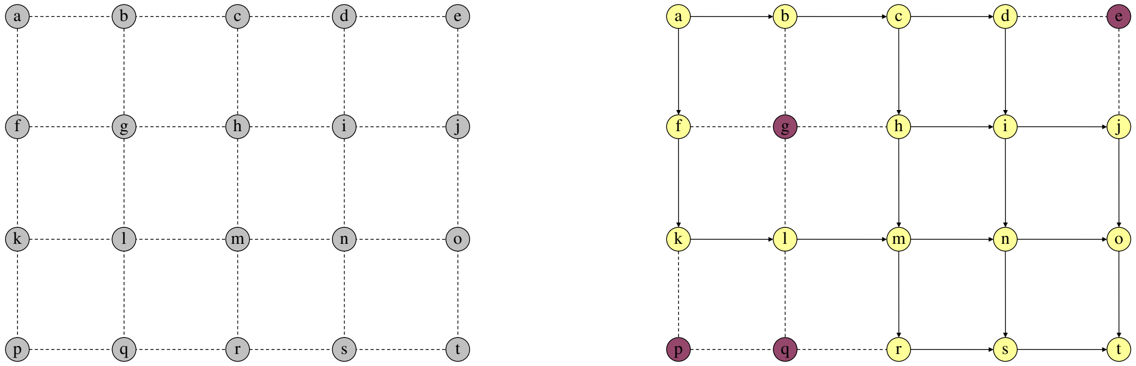

Fig. 3. The initially uninitialized network (left) where node *a* requires a route to node *t*. After route creation using the selective node participation approach, we obtain an initialized network (right) where the nodes assigned a *node_status* of active (yellow) participate as part of the network. Nodes that are assigned a *node_status* of inactive (maroon) do not participate thus reducing routing overhead. At the same time, there still exists multiple routes from node *a* to *t*.

Our modification differs from original TORA as nodes are assigned either an active, inactive, or unassigned status. All nodes are given an unassigned status when they first boot up. During route creation, a node propagates QRY packets when it requires a route to a destination. When a node *i* receives a QRY packet, it performs as follows:

1) If *node_status* is active, the QRY packet is processed as per original TORA.
2) If *node_status* is inactive, the QRY packet is ignored.
3) If *node_status* is unassigned and one of its neighbour is the destination or source, *node_status* is set as active and the QRY packet broadcast.
4) If *node_status* is unassigned and none of its neighbour is the destination or source, *node_status* is randomly set as active or inactive according to *probability_active* and the QRY packet is broadcast if the *node_status* assigned is active.

The flooding of QRY packets sets *node_status* in all nodes to either active or inactive. This reduces the QRY packets propagated compared to original TORA since nodes assigned an inactive status no longer propagate QRY packets. The optimal value for *probability_active* is dependent on the node density and mobility of the network. A high *probability_active* value makes it perform like original TORA while a low value creates unnecessary network partitions. After experimenting with various *probability_active* values, we found that a value of 0.8 works well in this case and used that in our simulations. Fig. 3 gives an example of how a network is initialized using the selective node participation approach.

Route maintenance and route erasure do not involve any nodes with a *node_status* of inactive. This does not decrease TORA's performance as a node *i* has many neighbours, and multiple routes exists through those neighbours. Network partitions are not unnecessarily created since those nodes do not participate in the DAG and hence there is no need to erase their heights.

V. Benchmark against Original TORA

In this section, we evaluate the performance of our proposed modifications of network localization and selective node participation against that of original TORA. For a comprehensive evaluation, we compare their performance using the metrics of packet delivery ratio, routing overhead, and average latency.

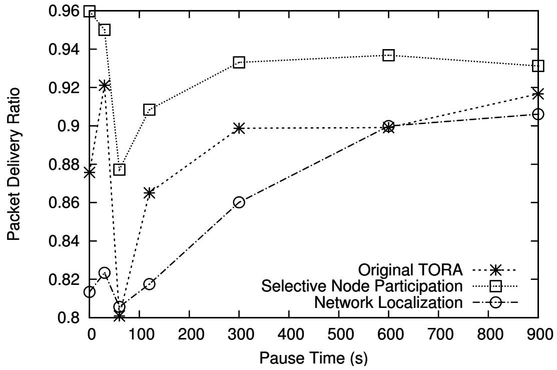

Fig. 4. Packet delivery ratio (50 nodes, 30 connections, 1 m/s)

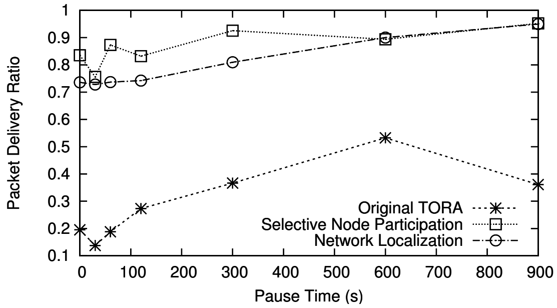

Fig. 5. Packet delivery ratio (100 nodes, 30, connections, 1 m/s)

A. Packet Delivery Ratio

Packet delivery ratio is defined as the total number of packets successfully delivered to their intended destination out of all packets sent. Fig. 5 shows that our two proposed modifications improved the packet delivery of TORA by up to 427% for large networks with many traffic connections at low mobility. We also obtained improvements in packet delivery for other network configurations with many traffic connections as shown in Fig. 4, 6 and 7.

Fig. 4, 5 and 6 show that the selective node participation approach performed better than the network localization ap-

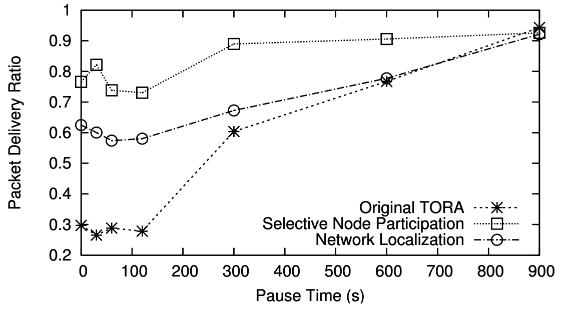

Fig. 6. Packet delivery ratio (50 nodes, 30 connections, 20 m/s)

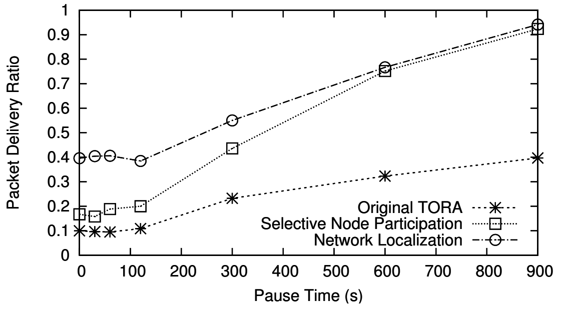

Fig. 7. Packet delivery ratio (100 nodes, 30 connections, 20 m/s)

proach for small networks at any mobility or large networks at low mobility, both with many traffic connections. However, the network localization approach performs better in large networks with many traffic connections at high mobility as shown in Fig. 7. We observe no significant packet delivery improvement for networks with few traffic connections.

While our proposed approaches perform better than original TORA, the amount of improvement depends on the network configuration and the chosen approach. The selective node participation approach initializes and maintains only certain routes (through selective nodes) that span the entire network. Thus, this approach performs better than the network localization approach for networks that are small or with low mobility. However, the network localization approach performs better for large networks with high mobility as it initializes and maintains routes only in a localized network segment containing the source and destination node.

B. Routing Overhead

Routing overhead allows us to study the number of routing packets required to facilitate the delivery of each data packet. Our proposed approaches of network localization and selective node participation reduces routing overhead by up to 95% and 80% respectively, as shown in Fig. 9. Similarly, Fig. 8, 10 and 11 show a reduction in routing overhead for small and large networks with many traffic connections at low and high mobility.

In terms of routing overhead, the network localization approach performs better compared to the selective node participation approach, especially for large networks. In large networks, the network localization approach initializes and maintains a much smaller subset of the entire network compared to the selective node participation approach, thus reducing more routing overhead in the process. This smaller subset of the entire network can be initialized and maintained as the source and destination are more likely to be near to each other instead of being at the extreme ends of the network.

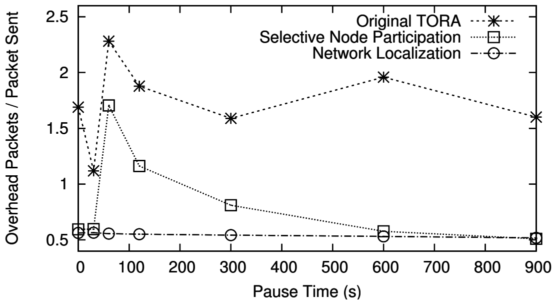

Fig. 8. Routing overhead (50 nodes, 30 connections, 1 m/s)

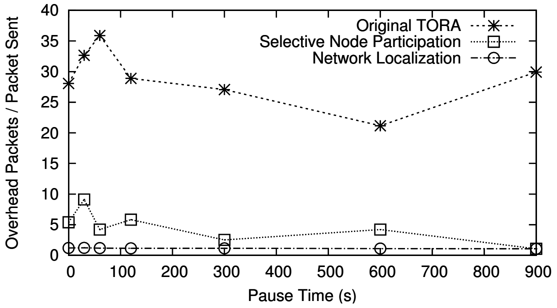

Fig. 9. Routing overhead (100 nodes, 30 connections, 1 m/s)

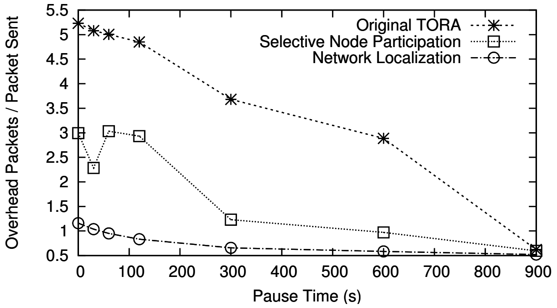

Fig. 10. Routing overhead (50 nodes, 30 connections, 20 m/s)

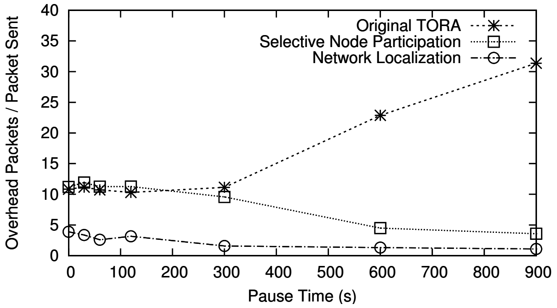

Fig. 11. Routing overhead (100 nodes, 30 connections, 20 m/s)

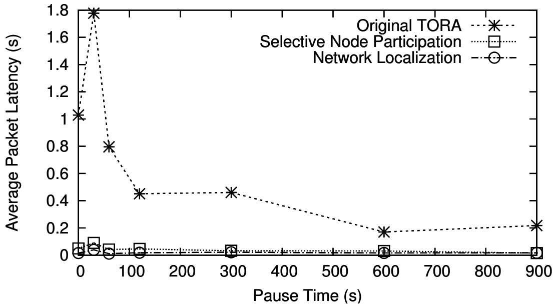

Fig. 12. Average latency (100 nodes, 30 connections, 1 m/s)

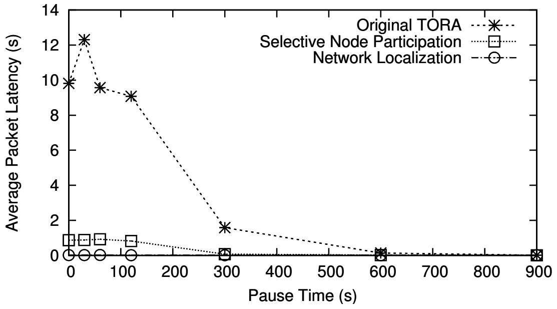

Fig. 13. Average latency (50 nodes, 30 connections, 20 m/s)

C. Average Latency

Average latency is defined as the time taken by a data packet to travel from a source node to its intended destination. Our proposed network localization and selective node participation approaches reduce latency by up to 99% and 91% respectively as shown in Fig. 13. Similarly, Fig. 12 and 14 shows a reduction in latency for large networks with many traffic connections at low and high mobility.

Our two proposed modifications enhance the performance of TORA by streamlining the route creation and maintenance functions of TORA. Route creation is first restricted to only a subset of the nodes (of the entire network) that are involved in communication. Thereafter, route maintenance is contained within this subset of nodes thus reducing the high routing overhead resulting from route maintenance. In turn, this alleviates the traffic congestion and allows data packets to be sent successfully. However, networks with little traffic connections do not suffer from high traffic congestion and thus gain no significant improvement from our proposed approaches. As such, our two proposed modification are most effective in networks with many traffic connections.

VI. Conclusion

We proposed two modifications to TORA: the network localization approach; and selective node participation approach. Specifically, the network localization approach initializes and maintains a DAG that represents a localized network (a portion of the entire network) while the selective node participation approach reduces the number of nodes that participate in the DAG but still maintain the overall integrity of the DAG. Both modifications are generally based on restricting the creation and maintenance of routes to a subset of the entire network. In turn, initializing and maintaining a subset of the entire network reduces excessive traffic congestion and allows for the delivery of more data packets in a timely manner. We also benchmarked our proposed modifications to original TORA and observed overall improvements in terms of an increase in packet delivery by up to 427% and reductions in routing overhead and packet latency by up to 95% and 99% respectively.

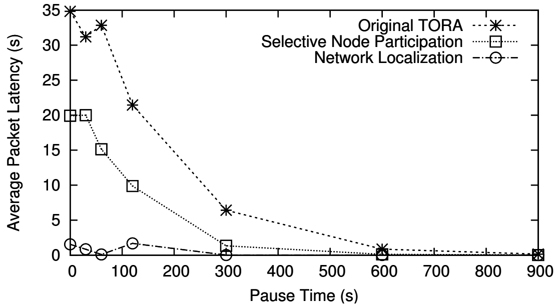

Fig. 14. Average latency (100 nodes, 30 connections, 20 m/s)

Acknowledgment

Kwan Hui Lim was supported by the Australian Government, University of Western Australia (UWA) and School of Computer Science and Software Engineering (CSSE) under the International Postgraduate Research Scholarship, Australian Postgraduate Award, UWA CSSE Ad-hoc Top-up Scholarship and UWA Safety Net Top-Up Scholarship.

References

[1] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, “A performance comparison of multi-hop wireless ad hoc network routing protocols,” in *Proc. of MOBICOM*, Oct 1998, pp. 85–97.

[2] I. Broustis, G. Jakllari, T. Repantis, and M. Molle, “A Comprehensive Comparison of Routing Protocols for Large-Scale Wireless MANETs,” in *Proc. of SECON*, Sep 2006, pp. 951–956.

[3] S. R. Das, R. Castaneda, and J. Yan, “Simulation-based Performance Evaluation of Routing Protocols for Mobile Ad hoc Networks,” *Mobile Networks and Applications*, vol. 5, no. 3, pp. 179–189, 2000.

[4] E. Weiss, G. R. Hiertz, and B. Xu., “Performance Analysis of Temporally Ordered Routing Algorithm based on IEEE 802.11a,” in *Proc. of VTC*, May 2005, pp. 2565–2569.

[5] V. D. Park and M. S. Corson, “A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks,” in *Proc. of INFOCOM*, Apr 1997, pp. 1405–1413.

[6] K. H. Lim and A. Datta, “An In-depth Analysis of the Effects of IMEP on TORA Protocol,” in *Proc. of WCNC*, Apr 2012, pp. 3051–3056.

[7] D. Dharmaraju, “Routing and Quality of Service in Mobile Ad Hoc Networks with TORA/INORA,” Master’s Thesis, University of Maryland, 2002.

[8] B. Coll-Perales and J. Gozalvez, “Energy efficient routing protocols for multi-hop cellular networks,” in *Proc. of PIMRC*, Sep 2009, pp. 1457–1461.

[9] M. S. Corson, S. Papademetriou, V. P. P. Papadopoulos, and A. Qayyum, “An Internet MANET Encapsulation Protocol (IMEP) Specification,” Internet Draft, Feb 1999, available from: http://www.tools.ietf.org/html/draft-ietf-manet-imep-spec-01.

[10] UCB, LBNL, and VINT, “The ns-2 Network Simulator,” Internet, Sep 2011, available from: http://nsnam.isi.edu/nsnam/index.php/Main_Page.

[11] B. Tuch, “Development of WaveLAN, an ISM band wireless LAN,” *AT&T Technical Journal*, vol. 72, no. 4, pp. 27–37, 1993.

[12] D. B. Johnson and D. A. Maltz, “Dynamic Source Routing in Ad Hoc Wireless Networks,” in *Mobile Computing*. Kluwer Academic Publishers, vol. 353.