An In-depth Analysis of the Effects of IMEP on TORA Protocol

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 highly adaptive distributed routing algorithm used in Mobile Ad hoc Networks (MANET) that provides multiple loop-free routes to a destination. TORA is very dependent on the services provided by the Internet MANET Encapsulation Protocol (IMEP) to carry out its three main functions: route creation, route maintenance, and route erasure. In this paper, we study how certain behaviour of IMEP leads to the poor performance of TORA and narrow down the problems of IMEP. We also proposed two modifications to IMEP that resulted in an overall performance improvement of TORA in terms of packet delivery, routing overhead and average packet latency. The original TORA and TORA based on our two IMEP modifications were also benchmarked against two other on-demand routing protocols: the Dynamic Source Routing (DSR) protocol and Ad-hoc On-demand Distance Vector (AODV) protocol.**

I. Introduction

A MANET comprises mobile nodes that are capable of wireless communication and movement in a random fashion. The connectivity between the nodes forms the topology of the MANET and one of its characteristic is a dynamic topology due to the unpredictable and sudden movement of nodes. TORA [1] is a highly adaptive distributed routing algorithm that is designed to work in such a network.

TORA performs well in terms of packet delivery for small networks with a small number of traffic connections but its performance decreases drastically with a large number of traffic connections [2], [3]. Broch *et al.* and Broustis *et al.* [2], [3] attributed this problem to certain characteristics of IMEP and how TORA contributes to these characteristics. While Park and Corson [1] showed that TORA provides loop-free routes, temporary loops may still exist during the maintenance of these routes. For example, Weiss *et al.* [4] pointed out that loops could be formed when control messages are not immediately delivered. Similarly, Das *et al.* [5] noted the possibility of short-lived inconsistencies about the direction of links that lead to loops.

Dharmaraju [6] proposed a *Query Localization* approach to modify and improve the performance of TORA. This approach attempts to reduce routing overhead by querying routes in an expanding ring manner instead of flooding the entire network. At the time of writing this paper, there has been no prior study on the effects of IMEP on TORA nor has there been any attempt to enhance the performance of TORA via IMEP. Our contributions include a study into how certain services of IMEP affect the performance of TORA and two modifications to IMEP that improve the overall performance of TORA.

In this paper, we introduce how TORA works in Section II and analyze IMEP in detail, particularly in the way it detects link failures, in Section III. Next, we give an overview of our approach to the problem in Section IV. In Section V, we suggest modifications to the existing IMEP protocol and evaluate how original TORA performs against TORA using our IMEP modifications. In Section VI, benchmarks are performed against the DSR and AODV protocols. Finally, we conclude the paper in Section VII by summarizing the key points.

II. Temporally-Ordered Routing Algorithm

TORA is a distributed routing algorithm based on link reversals. TORA provides multiple loop-free routes to any destination on-demand by modelling the entire network as a directed acyclic graph (DAG). Each node has an associated height, and links between nodes flow from a higher height to a lower one. This collection of links forms the DAG.

TORA operates using three main functions of route creation, maintenance and erasure. Route creation establishes the DAG and provides all nodes with a route to a particular destination while route maintenance maintains the integrity of the DAG. Route erasure removes all invalid routes when a node detects that it is in a network partition with no route to the destination. In addition, TORA uses various IMEP services to perform these three functions. More information on the TORA protocol can be found in the original paper by Park and Corson [1].

III. Internet MANET Encapsulation Protocol

IMEP provides services that TORA requires such as link/connection status sensing, broadcast reliability, and message aggregation. With reference to the Open Systems Interconnection (OSI) Model [7], IMEP sits below TORA with both protocols residing at the network layer. In turn, IMEP uses the services provided by the Institute of Electrical and Electronics Engineers (IEEE) 802.11 Medium Access Control (MAC) protocol [8], a data link layer protocol. The following description of IMEP is based on the internet draft of IMEP developed by Corson *et al.* [9].

A. Message Aggregation

Message aggregation encapsulates IMEP's own routing control packets and packets passed down by TORA into a single object block message (OBM). This minimizes the number of channel accesses needed since a single OBM packet is sent instead of multiple, smaller IMEP and TORA packets.

B. Link/Connection Status Sensing

Link/connection status sensing provides TORA with accurate and current link status information of a node to its neighbours and whether the links are bi-directional or uni-directional. It operates using the explicit and implicit method of detecting link failures.

1) Explicit Link Failure Detection Method: This method determines link status information by having a node i broadcast BEACON packets to its one-hop neighbours. When node i receives a reply in the form of an ECHO packet from a neighbour, it labels the link to that neighbour as bi-directional. Node i continues to send BEACON packets at every BEACON_PERIOD interval and sets a MAX_BEACON_TIME, derived from the maximum number of BEACON retransmissions multiplied by the BEACON_PERIOD. If node i does not hear any ECHO packet after MAX_BEACON_TIME, it labels that link as down and notifies TORA.

2) Implicit Link Failure Detection Method: This method utilizes the OBM packets that IMEP sends, where nodes who receive an OBM packet reply with an ACK packet. This procedure mirrors that of the BEACON and ECHO packets used in the explicit method discussed earlier. Similarly, there is a MAX_RETRANS_TIME made up of the RETRANS_PERIOD between each retransmission multiplied by the maximum number of retransmissions. After MAX_RETRANS_TIME, any neighbour that did not reply with an ACK packet has its link labelled as down and TORA will be notified.

C. Broadcast Reliability

Broadcast reliability can be any mix of two delivery modes of broadcast or multicast, and also two reliability modes of reliable or unreliable. Specifically, TORA requires broadcast reliability in the reliable and broadcast mode, ensuring in-sequence delivery of messages and broadcasting to all of its neighbouring nodes. The broadcast mode requires all neighbouring nodes to acknowledge any OBM packet sent and this facilitates link/connection status sensing in the implicit method of link failure detection.

IV. Experimental Approach

Incorrect detection of link failures causes TORA to perform unnecessary route maintenance, resulting in a congested network and data packets not reaching their destinations. Hence, we have to examine the link/connection status sensing service provided by IMEP for the detection of link failures.

One way that IMEP can incorrectly detect a link failure is when it does not receive ECHO packets in response to a BEACON packet sent. This is the explicit method of link failure detection which we first examine and show that it contributes minimally to the incorrect detection of link failures.

Next, we show that the implicit method of link failure detection is the main cause of the incorrect detection of link failures. The main problem is due to ACK packets failing to reach the senders of OBM packets causing the senders to think that their neighbours are down.

TABLE I
IMEP constants

IMEP Constant	Value
BEACON_PERIOD	1 s
Max. no. of BEACON retransmissions	3 times
MAX_BEACON_TIME	3 s
Object block message RETRANS_PERIOD	0.5 s
Max. no. of object block message retransmissions	3 times
MAX_RETRANS_TIME	3 s
Min. message aggregation delay	150 ms
Max. message aggregation delay	250 ms

V. Experiments and Results

Broch *et al.* and Broustis *et al.* [2], [3] noted that the loss of data, ACK and BEACON packets due to heavy traffic causes IMEP to incorrectly detect link failures. There has been no study that shows how this error occurs or its impact on TORA. We examine this and suggest modifications to IMEP that improves the performance of TORA.

A. Methodology

We used the *ns-2* network simulator [10], a discrete event simulator that provides a controlled environment where we can examine the characteristics of IMEP and TORA. The radio propagation properties of mobile nodes are modelled after the Lucent WaveLan direct sequence spread spectrum radio [11]. The random waypoint model [2] is used to define movement scenarios for nodes in the network. As IMEP [9] does not specify the values for the constants used, we chose the values used by Broch *et al.* [2] as shown in Table I.

We modelled nodes with a maximum speed of 1 m/s or 20 m/s, representing a low and high mobility network respectively. The networks of 50 and 100 nodes were designated a topology size of $1500m \times 300m$ and $2121m \times 425m$ respectively, maintaining a node density of approximately $9000m^2$ per node. The pause times chosen were 0, 30, 60, 120, 300, 600, and 900 seconds where 0 and 900 seconds represents constant mobility and no movement respectively.

Traffic connections in the network are modelled using constant bit rate (CBR) sources where packets are sent to a particular destination at a constant rate until the end of the simulation. We used CBR packets with a size of 64 bytes and sending rate of 4 packets per second.

As the simulation results obtained are very sensitive to the movement scenarios used, we simulated the 50 and 100 nodes network using 20 and 10 different movement scenarios respectively. The results obtained were then averaged and this process repeated for each unique combination of pause time, number of traffic connections, and node mobility.

B. Analysis of Explicit Link Failure Detection

In a network with many traffic connections, IMEP is able to detect the status of links implicitly as many OBM packets are sent frequently. We prove this by showing that the explicit method of link failure detection affects TORA minimally.

1) Increasing the Maximum Number of BEACON Retransmissions: If incorrect link failure detection is caused by the inability of BEACON packets to reach nodes, modifying IMEP by increasing the maximum number of BEACON retransmissions solves the problem. This modification also increases MAX_BEACON_TIME thus the probability that nodes receive BEACON packets and subsequently reply with ECHO packets. However, our results show that this modification does not significantly improve TORA, regardless of network size, node mobility or the number of traffic connections. Improvements of up to 80% can be observed for large networks with 30 traffic connections but only when the network is static.

2) Extending BEACON_PERIOD: This method tests if nodes receive the BEACON packets but are unable to access the physical medium (due to heavy traffic) to reply with ECHO packets. If true, extending the BEACON_PERIOD provides the nodes with a longer timeframe to reply. We observed that extending the BEACON_PERIOD does not bring about any significant performance improvement, as is the case with increasing the maximum number of BEACON retransmissions.

Both modifications of increasing the maximum number of BEACON retransmissions and extending the BEACON_PERIOD do not improve the performance of TORA significantly. This lack of improvement shows that the explicit method of link failure detection has little effect on the performance of TORA and is not the main cause of incorrect link failure detection. In a network with many traffic connections, OBM packets are frequently sent thus the implicit method of link failure detection takes precedence and the explicit method becomes redundant. If there is little traffic, nodes are able to send and receive BEACON and ECHO packets successfully since there is little chance of lost packets due to network congestions. Hence, the explicit method of link failure detection only plays an important role during route creation. After which, actual traffic (i.e. OBM packets) starts and IMEP determines link status using the implicit method.

C. Analysis of Implicit Link Failure Detection

ACK packets that fail to reach the senders of OBM packets cause incorrect link failure detection by the implicit method. This can happen in two ways: neighbouring nodes receive OBM packets but are unable to reply with ACK packets due to a congested physical medium, or OBM packets failing to reach neighbouring nodes in the first place.

Excessive overhead packets are caused by the broadcast and reliable mode of IMEP required by TORA. Broadcasting an OBM packet requires all neighbouring nodes to reply with ACK packets, generating excessive traffic that increases with the number of traffic connections.

1) Increasing the Maximum Number of OBM Retransmissions: Adapting a similar modification as the explicit method of link failure detection, we increase the maximum number of OBM retransmissions to five times. This modification increases the probability that an OBM packet sent by a node

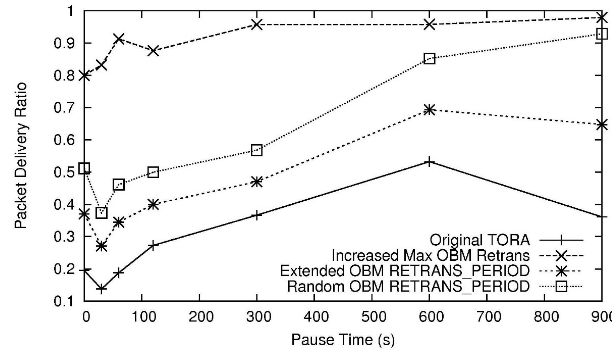

Fig. 1. Implicit method (100 nodes, 30 connections, 1 m/s)

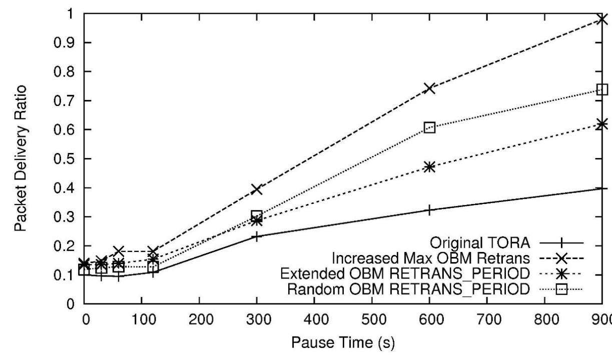

Fig. 2. Implicit method (100 nodes, 30 connections, 20 m/s)

reaches all of its neighbouring nodes. Nodes that fail to reply with ACK packets within MAX_RETRANS_TIME have their links labelled as down. This is the broadcast mode required by TORA and it increases the chance that IMEP incorrectly detects a link breakage. We now discuss the results obtained by modifying the implicit method of link failure detection.

The results we obtained support our claim that the implicit method of link failure detection is the main cause of the incorrect detection of link breakages. This modification does not decrease the performance for networks with few traffic connections. On the other hand, Fig. 1 shows that TORA obtains performance improvement of 400% in a network of 100 nodes moving constantly at 1 m/s with 30 traffic connections. Even for a network of 100 nodes with a mobility of 20 m/s and 30 traffic connections, a performance improvement of 12% and 246% is observed for constant and no mobility respectively, as shown in Fig. 2. This disparity in performance improvement of 12% to 400% is because in high mobility networks, many actual link breakages occur and are reported (correctly) by IMEP. However, in low mobility networks, link breakages seldom occur but are still frequently reported (incorrectly) by IMEP. Hence, the performance improvement gained from correcting these detection errors is more significant in low mobility networks than high mobility ones.

The underlying reason for this performance increase is that in the original protocol, neighbouring nodes fail to receive OBM packets and thus do not reply with ACK packets within MAX_RETRANS_TIME. The links to these neighbouring nodes are then labelled as down and TORA initiates route maintenance thus creating even more overhead. Increasing the maximum number of OBM retransmissions creates more chances for OBM packets to reach neighbouring nodes. These nodes can then reply with ACK packets thus reducing the amount of link failures that are incorrectly detected.

In the IEEE 802.11 MAC protocol, broadcasting utilizes virtual and physical carrier sensing but not Request-To-Send and Clear-To-Send packets; nor does it require receiving nodes to acknowledge the reception of packets [8]. Hence, when IMEP sends packets down to the MAC layer for broadcasting, there is no indication whether the packets reached its recipients or collided in the transmission medium. Increasing the maximum number of OBM retransmissions improves the chances of a packet reaching its destination even if the first few packets collided upon transmission.

2) Extending OBM RETRANS_PERIOD: It is also important to determine if the OBM packet reached the neighbouring nodes but they were unable to reply with an ACK packet in time due to a busy medium. If that is the case, extending the RETRANS_PERIOD solves the problem. Fig. 1 shows an improvement of about 100% compared to 400% gained by increasing the maximum number of OBM retransmissions, which is still the main factor in improving IMEP.

3) An Approach based on Random OBM RETRANS_PERIOD: This approach reduces the probability of multiple nodes retransmitting concurrently following a packet collision. IMEP is modified such that each node is assigned a random OBM RETRANS_PERIOD between 0.5 and 0.8 seconds, instead of a fixed interval.

Fig. 1 and 2 show that this approach increases the overall performance of TORA with a large number of traffic connections. Of particular interest is the higher performance increase of approximately 250% compared to 100% by increasing the RETRANS_PERIOD. However, this improvement is less than the 400% improvement gained by increasing the maximum number of OBM retransmissions.

Increasing the maximum number of OBM retransmissions produces the largest improvement in TORA, followed by the random OBM RETRANS_PERIOD approach, then extending the RETRANS_PERIOD. The first modification performing the best shows that neighbouring nodes are unable to receive OBM packets and thus do not reply with ACK packets. The improvements from modifying the implicit method of link failure detection further reinforces our original claim that in a network with many traffic connections, the implicit method takes precedence and the explicit method is hardly utilized.

VI. Benchmark against other Protocols

The DSR [12] and AODV [13] on-demand protocols are chosen as benchmarks to the TORA protocol. DSR is based on source routing where packets contain the entire route to reach a given destination as part of its packet header. On the other hand, AODV uses hop-by-hop routing where nodes maintain only the next hop to forward a packet, for the packet

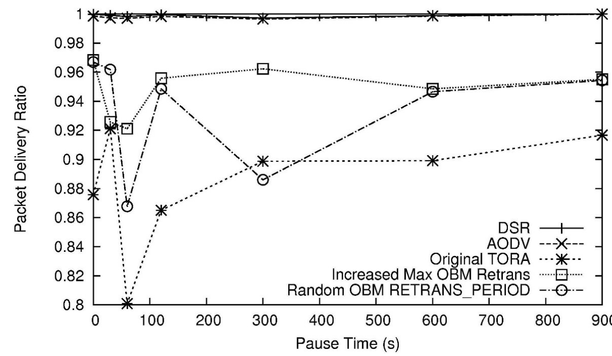

Fig. 3. Packet delivery ratio (50 nodes, 30 connections, 1 m/s)

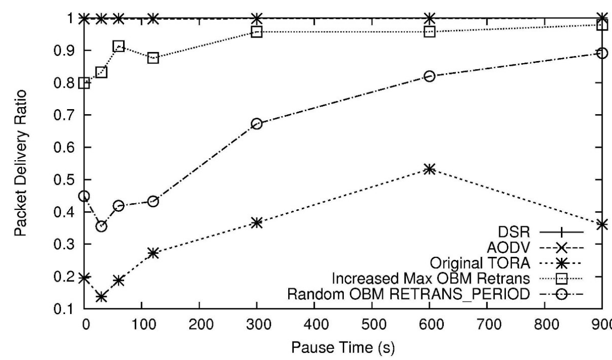

Fig. 4. Packet delivery ratio (100 nodes, 30, connections, 1 m/s)

to reach a given destination. Refer to [12] and [13] for a detailed description of DSR and AODV respectively.

We now compare the performance of original TORA and TORA based on our two best performing IMEP modifications against DSR [12] and AODV [13]. The two IMEP modifications that we compare are that of increasing the maximum number of OBM retransmissions, and the approach based on a random OBM RETRANS_PERIOD. We evaluate their performance using packet delivery ratio, routing overhead, and average latency.

A. Packet Delivery Ratio

Packet delivery ratio is derived from the total number of data packets received out of the total number of data packets sent. TORA does not perform as well as DSR and AODV but we obtained improvements in certain scenarios that brings it close to DSR and AODV, as shown in Fig. 3. Other examples are shown in Fig. 4 and 5 where we increase the packet delivery ratio of TORA to 0.8, or just 20% less than that of DSR and AODV. No significant improvement is observed for small and large networks with 10 traffic connections at 1m/s and 20m/s mobility.

Fig. 5 and 6 show that our IMEP modifications work better for small networks than large ones, with many traffic connections and high mobility. This is because a large network requires the maintenance of a larger DAG than that of a small network. Even after correcting the incorrect detection of link failures, a large DAG still incurs more routing overhead than a

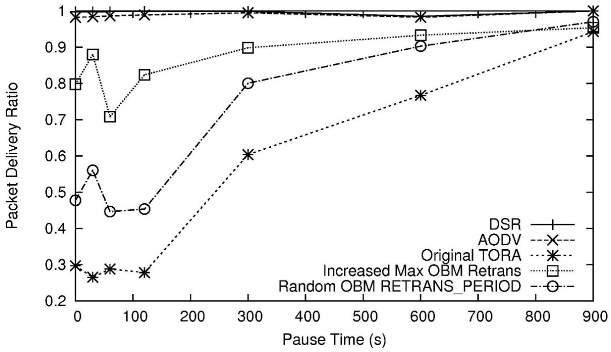

Fig. 5. Packet delivery ratio (50 nodes, 30 connections, 20 m/s)

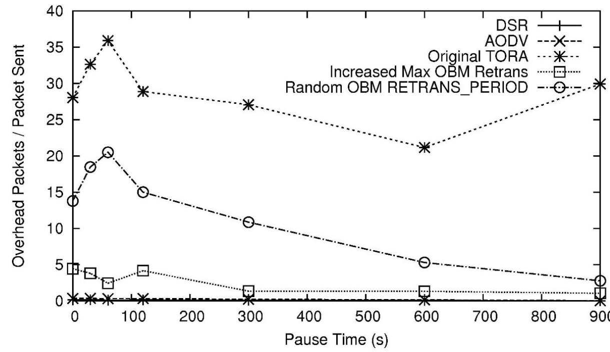

Fig. 7. Routing overhead (100 nodes, 30 connections, 1 m/s)

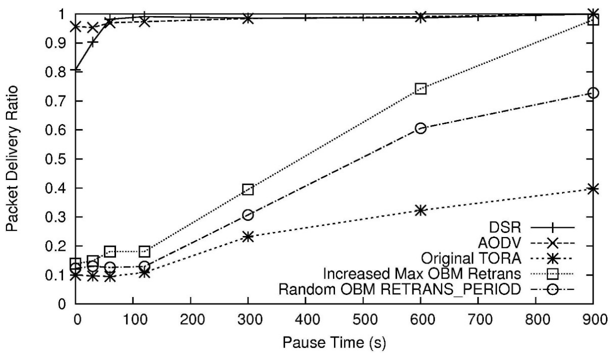

Fig. 6. Packet delivery ratio (100 nodes, 30 connections, 20 m/s)

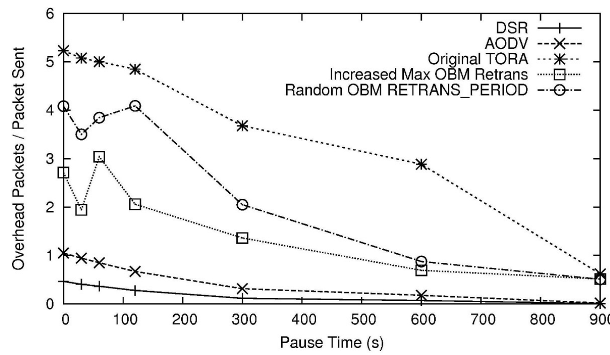

Fig. 8. Routing overhead (50 nodes, 30 connections, 20 m/s)

small DAG as many detections are actual link failures. These additional routing overheads thus prevent data packets from reaching their destinations.

B. Routing Overhead

We measure routing overhead in terms of the number of routing packets required for each data packet sent. The routing overhead graphs reflect the trend of the corresponding packet delivery ratio graphs such that an increase in packet delivery corresponds to a decrease in routing overhead. These two sets of results support our initial claim that excessive routing overhead hinders data packets from reaching their destinations.

Fig. 7 and 8 show up to 84% and 48% reduction in routing overhead respectively, when comparing TORA using our two IMEP modifications against original TORA. Comparing the two IMEP modifications, the approach by increasing the maximum number of OBM retransmissions generates less routing overhead than that of using a random OBM RETRANS_PERIOD. We observed less reduction in routing overhead for other network configurations. While our IMEP modifications reduce routing overhead, DSR and AODV produce less overhead comparatively.

C. Average Latency

Average latency allows us to observe the time a packet takes to reach its destination from the source node. Average packet latency does not improve significantly for a low mobility network with 50 nodes but more so when the network size increases to 100 nodes. Fig. 9, 10, and 11 show that original TORA has the highest average packet latency, while our approaches of increasing the maximum number of OBM retransmissions, and using a random OBM RETRANS_PERIOD reduce latency by up to 91% and 55% respectively. This reduction in latency makes TORA comparable to DSR and AODV for certain scenarios such as in Fig. 9 and 10.

Fig. 12 shows that for original TORA, there are a few outliers in the form of packets that take between 64 and 512 seconds to reach their destinations. These outliers are the main cause of a high average packet latency of 31 seconds as shown in Fig. 11, at a pause time of 30 seconds. These packets belong to intermediate nodes that cannot find a route to the destination due to the incorrect detection of link failures by IMEP. These nodes then store the packets in a queue for an extended period until a route is found, contributing to the long latency.

Our IMEP modifications reduce latency by up to 91%, 90% and 34% as shown in Fig. 9, 10 and 11 respectively. Even with a pause time of 30 seconds, Fig. 11 shows that our two approaches of increasing the maximum number of OBM retransmissions and using a random OBM RETRANS_PERIOD reduce the average packet latency by 31% and 24% respectively. Fig. 12 best explains this by showing that while our approaches do not reduce the number of outliers (i.e. packets with latency between 64 and 512 seconds), it increases the number of packets delivered within 1 second. This is partly due to the improvement in packet delivery ratio which ensures that more packets can be delivered within 1 second.

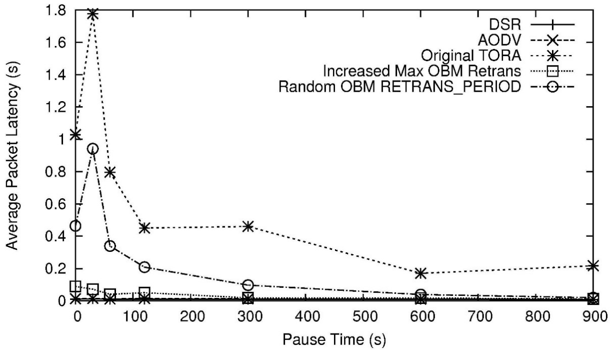

Fig. 9. Average latency (100 nodes, 30 connections, 1 m/s)

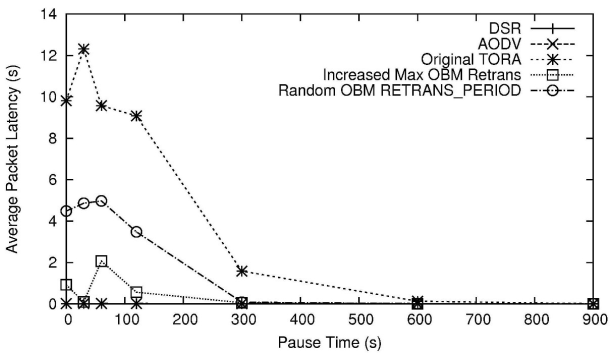

Fig. 10. Average latency (50 nodes, 30 connections, 20 m/s)

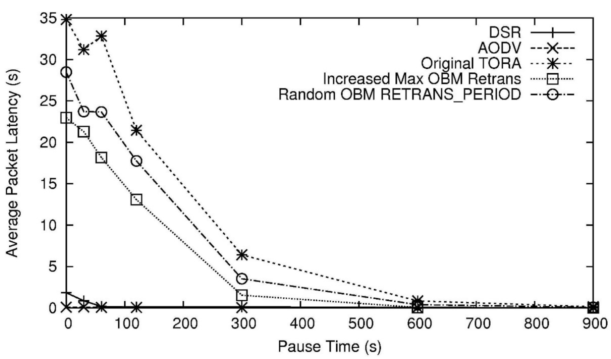

Fig. 11. Average latency (100 nodes, 30 connections, 20 m/s)

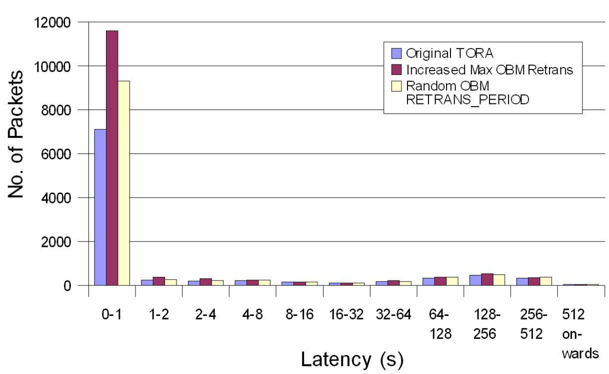

Fig. 12. Breakdown of packet latency for 100 nodes with 30 traffic connections at 20 m/s and pause time of 30 s

VII. Conclusion

In this paper, we studied how IMEP affects the performance of TORA with the link/connection status sensing service being the main factor. Upon closer examination, we found that this service is very dependent on the maximum number of OBM retransmissions for the implicit method of link failure detection. On the contrary, the explicit method of link failure detection plays a minimal role. We also presented two modifications to IMEP using a random OBM RETRANS_PERIOD, and increasing the maximum number of OBM retransmissions. Both modifications resulted in an overall improvement of TORA in terms of packet delivery, routing overhead and average latency for various network configurations.

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] V. D. Park and M. S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," in *Proc. of INFOCOM '97*, Apr 1997, pp. 1405–1413.

[2] 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 the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, Oct 1998, pp. 85–97.

[3] I. Broustis, G. Jakllari, T. Repantis, and M. Molle, "A Comprehensive Comparison of Routing Protocols for Large-Scale Wireless MANETs," in *3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, Sep 2006, pp. 951–956.

[4] E. Weiss, G. R. Hiertz, and B. Xu., "Performance Analysis of Temporally Ordered Routing Algorithm based on IEEE 802.11a," in *Proc. of the 61st IEEE Vehicular Technology Conference*, May 2005, pp. 2565– 2569.

[5] 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.

[6] D. Dharmaraju, "Routing and Quality of Service in Mobile Ad Hoc Networks with TORA/INORA," MSc Thesis, Uni. of Maryland, 2002.

[7] J. D. Day and H. Zimmermann, "The OSI Reference Model," in *Proc. of the IEEE*, vol. 71, Dec 1983, pp. 1334– 1340.

[8] IEEE Computer Society LAN MAN Standards Committee, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," Standard, 2007. [Online]. Available: http://standards.ieee.org/getieee802/download/802.11-2007.pdf

[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.

[13] C. E. Perkins and E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing," in *Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, Feb 1999, pp. 90–100.