The Impact of Initial Network Topology on Performance of Routing Algorithms in MANETs

Ronit Nossenson and Adi Schwartz

***Abstract*— The impact of the initial network topology on performance of routing algorithms is explored. Typically researchers use a randomly chosen network topology for performance evaluation of their protocols and algorithms. Here we show that the initial network topology can have a significant impact on algorithm performance and can lead to biased results, in particular, an initial topology that includes a major connectivity obstacle such as low connectivity level (e.g., a tree topology) or bridges. Although users move according to commonly implemented random mobility models, the effect of the initial topology can persist over time. To avoid biased results we recommend using multiple initial topologies instead of one, and/or running the simulation in an initialization phase until the effect of the initial topology fades.**

***Index Terms*— MANET, routing algorithms, performance evaluation, network topology**

I. INTRODUCTION

IN wireless multi-hop ad-hoc networks each node not only acts as a possible source and sink of traffic, but also as a router enabling the forwarding of traffic between nodes in the network. Mobile routing protocols in such networks must be efficient and must either keep their knowledge proactively up to date, or be highly reactive when routes are required. Over the past decade, hundreds of different routing protocols have been proposed for wireless ad-hoc networks.

The performance evaluation of routing algorithm is frequently carried out via a simulation on a randomly generated ad-hoc network topology. In addition, a mobility model is assumed to mimic the users' mobility. Researchers usually assume that using a randomly generated network topology with proper mobility is an objective environment to study the behavior of their protocols and algorithms. In this report we show that this common assumption is flawed and demonstrate that a connectivity obstacle in the initial network topology can bias performance results.

Previous works have studied the performance of routing protocols in ad-hoc networks; see for example [2, 7, 8, 9, 10, 11] but none have investigated the impact of the initial network topology.

To understand the impact of the initial topology on performance we extended the Network Simulation 2 (NS2) [1] with a topology generator tool. The topology generator creates a network topology graph of nodes according to given prerequisites that makes it possible to create specific characteristic topologies such as a tree topology, a graph with a bridge edge, a well connected graph, etc.

Using the topology generation tool we generated three types of network topologies starting from a tree type having exactly one path between every two nodes, up to a well connected topology graph. We used three mobility scenarios: no mobility, low mobility and medium mobility. We tested the performance of four well known routing algorithms on these topologies and compared the results. The four algorithms were: Dynamic Source Routing (DSR) [3], Ad -hoc On-Demand Distance Vector (AODV) [4], Destination-Sequenced Distance Vector Routing (DSDV) [5], and Ad–Hoc On-demand Multipath Distance Vector (AOMDV) [6].

The first very striking finding was that a connectivity obstacle in the initial topology of the network resulted in lower performance results of all the algorithms we tested. Furthermore, the initial topology can bias the comparative results across algorithms. Furthermore, the impact of user mobility on algorithm performance varies as a function of the initial topology. In a well -connected initial topology, the algorithm performance decreases with increased user mobility whereas in an initial topology with connectivity obstacles performance can improve when user mobility increases.

The direct implication of these findings is that researchers must be aware of the fact that their randomly generated initial topology can bias performance results. To avoid this pitfall, we suggest using more than one random initial topology (that is, repeat the simulation several times with different initial topologies), and run the simulation in an initialization phase before starting to collect the statistics.

This paper is organized as follows. In the next section we describe the four routing algorithms. Section III consists of the simulation description. The results are presented in Section IV. Finally, the implications are discussed in Section V.

II. ROUTING PROTOCOLS

There are two main categories of protocols in ad hoc routing: proactive and reactive. Proactive routing protocols are based on 'normal' routing protocols used in wired networks, such as today's Internet. Algorithms based on

---

Manuscript received April 10, 2012; revised March 20, 2012.

R. Nossenson is with the Jerusalem College of Technology (JCT), Jerusalem, Israel (corresponding author: 972-9-7679272; fax: 972-9-7679272; e-mail: nossenso@ jct.ac.il).

A. Adi Schwartz is with The Academic College of Tel-Aviv, Tel-Aviv, Israel (e-mail: adishh@gmail.com).

distance vectors or link states are commonplace. In proactive protocols, a local representation of routes at each node within the network is built before they are put into use. The routing tables are usually built up periodically through normal operation of the protocol by exchanging routing update packets. In normal operation, this has the advantage that the routes are already pre-computed and packet forwarding can take place as soon as a packet for a particular destination appears at a node. The drawback is that routes may be calculated and re-calculated (for example due to node mobility) when they are not actually required. This wastes bandwidth and, for mobile nodes, also wastes battery power because unnecessary routing updates are sent and received.

Reactive routing takes an alternative approach by building routes solely on demand. It can also cache route information according to some short time-out or staleness policy. Cached routes can be used as required, but if a route is not known then it has to be ‘discovered’. This has the advantage that routes are only evaluated when needed, although this approach adds latency to packet forwarding when routes are not already known.

A. *Dynamic Source Routing (DSR) [3]*

The DSR routing algorithm is a reactive routing algorithm that uses source routing; in other words, the sender knows the complete route to the destination. These routes are stored in a route cache. The data packets carry the source route in the packet header. When a node in the ad hoc network attempts to send a data packet to a destination for which it does not already know the route, it uses a route discovery process to dynamically determine it. Route discovery works by flooding the network with route request (RREQ) packets. Each node receiving an RREQ rebroadcasts it, unless it is the destination or it has a route to the destination in its route cache. This node then replies to the RREQ with a route reply (RREP) packet that is routed back to the original source. If any link on a source route is broken, the source node is notified using a route error (RERR) packet and the route is removed from the cache.

B. *Ad -hoc On-Demand Distance Vector (AODV) [4]*

The AODV routing algorithm is a reactive routing algorithm. It discovers routes on an as -needed basis via a route discovery mechanism. It uses traditional routing tables, with one entry per destination. AODV relies on routing table entries to propagate an RREP back to the source and to route data packets to the destination. AODV uses sequence numbers maintained at each destination to determine the freshness of routing information and to prevent routing loops. All routing packets carry these sequence numbers.

A routing table entry expires if not used recently. In addition, when a failure occurs, RERR packets in AODV are used to inform all sources using this link.

C. *Destination Sequenced Distance Vector (DSDV) [5]*

The DSDV routing algorithm is a proactive routing algorithm. It is based on the Bellman-Ford routing algorithm with certain improvements. Every mobile station maintains a routing table that lists all available destinations, the number of hops to reach the destination and the sequence number assigned by the destination node. The sequence number is used to distinguish stale routes from new ones and thus avoid the formation of loops. The stations periodically transmit their routing tables to their immediate neighbors. A station also transmits its routing table if a significant change has occurred. Thus the update is both time-driven and event-driven. The routing table updates can be sent in two ways: full or incremental.

D. *Ad –Hoc On-demand Multipath Distance Vector (AOMDV) [6]*

The AOMDV is a reactive routing algorithm. It extends AODV to discover multiple link-disjoint paths between the source and the destination in every route discovery. It uses the routing information already available in the AODV protocol as much as possible. It makes use of AODV control packets with a few extra fields in the packet header such as advertised hop count and a route list which contains multiple paths.

III. SIMULATION DESCRIPTION

The simulation was performed using the NS2 simulator version 2.34 [1]. We compared the DSR [3], AODV [4], DSDV [5], and AOMDV [6] routing algorithms under different initial topologies and mobility scenarios.

A. *The simulation process*

The simulation process consisted of the following steps as shown in Figure 1.

1) The topology generator creates a network topology graph of nodes according to given prerequisites that create specific characteristic topologies (described in the next sub- section).
2) The mobility generator uses the CANU mobility simulator [9]. It creates user profiles: static, slow walking and vehicular movement.
3) The traffic generator creates different traffic types. It

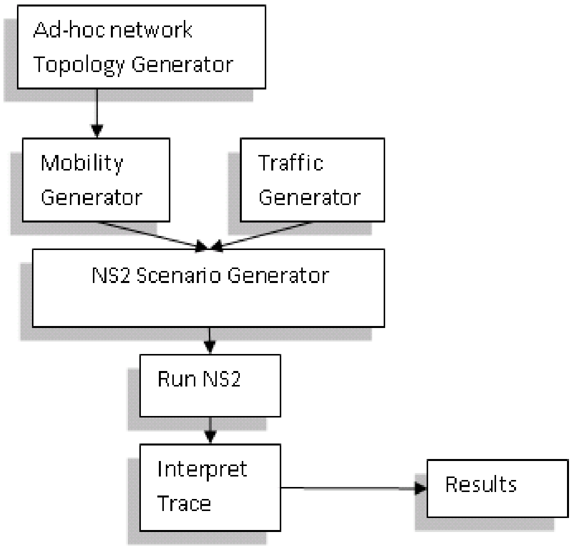

Fig. 1: The simulation process

creates VoIP traffic, using Constant Bit Rate, with small packets transmitted over UDP. Web-browsing traffic is created using an ON/OFF process with a Pareto distribution, with large packets transmitted over TCP. In addition, it creates video traffic using a smoother Pareto ON/OFF process with large packets transmitted over TCP.

4) The NS2 scenario generator composites the mobility and traffic generated data into a specific ns2 scenario template. The ns2 scenario template runs the same scenario with the same initial network topology on different routing algorithm, leaving the routing algorithm as a parameter.
5) The trace interpreter reads the NS2 generated trace file and extracts performance related information such as throughput, average drop rate and average packet delay.

B. *Topology Characteristics*

To study the impact of the initial network topology we extended the NS2 to include a topology generator tool. The topology generator generates a random ad-hoc network topology based on the prerequisite list below. Each of the tested topologies was used as a starting point for the mobile nodes (before they started moving according to the mobility scenario). We created ten different instances of each topology type. All the generated topologies had the same number of nodes and were located in the same area size.

This random topology generator function can create the following three types of characterized topologies:

1) A tree
2) A bridged graph, having a link such that removing it decuples the graph into two connected graphs
3) A well connected graph having two disjoint paths from each node to each node

C. *Traffic Generation*

The simulator modeled three different traffic generator functions:

1) The VoIP traffic generator was a Constant Bit Rate application with a 100 kbps connection based on a UDP connection.
2) The Web browsing traffic generator was a Pareto application over TCP with a packet size of 1000 bytes, an average burst time (ONN period) of 100msec, an average idle time (OFF period) of 700msec, a rate of 500 kbps and a shape parameter (alpha) of 1.1
3) The video traffic generator was also a Pareto application with a shape parameter of 2.2, an ON period average of 500msec, an average idle time (OFF period) of 100msec and a rate of 500kbps.

We used a mixed traffic type scenario with 1/3 of the nodes generating VoIP traffic, 1/6 generating web browsing traffic and another 1/6 generating video- like traffic (the other nodes do not generate traffic).

D. *The Mobility Generator*

The mobility generator takes the topology area that was produced by the topology generator and generates a mobility scenario using the CANU mobility simulator [12]. There are three different mobility profiles for a given user:

1) A static user: without any movement
2) A random walking user with a speed of 1-5kmh and a 5s-50s pause duration
3) A random driving user with a speed of 30-120kmh and a 0s-25s pause duration

We considered three mobility scenarios in our simulation: (i) a no-mobility scenario, (ii) a low-mobility scenario in which 2/3 of the users are static and 1/3 have a random walking profile; and (iii) a medium-mobility scenario in which 1/3 of the users are static, 1/3 have a random walking profile and 1/3 have a random driving profile.

E. *The Network interface*

The network interface was set to wireless PHY with a propagation distance of 100m and a radio propagation model of Two Ray Ground with an Omni antenna. The Queue is a priority queue that gives a higher priority to routing protocol packets, the queue size was set to 50 and the MAC was based on the 802.11 wireless LAN.

IV. RESULTS

The key simulation results are presented in this section. The impact of the initial network topology on the packet drop percentages is presented in Figure 2-5. As expected, initial topologies of weakly connected graphs such as the tree-graph and the bridge-graph increase the packet drop percentages in all algorithms. The DSR outperformed the other algorithms and maintained low packet drop percentages.

Strikingly, increasing the mobility level of the users had a differential impact on the drop percentages as a function of the initial network topology. For initial network topologies that are well connected, the increase in user mobility generally resulted in an increase in the drop percentage. For example in the DSDV algorithm (Figure 3), increasing the mobility level from low to medium with a well connected initial topology resulted in drop percentages of 33% instead of 14%. By contrast, for initial network topologies with connectivity obstacles, the reverse was true: the increase in user mobility from low to medium mobility resulted in a decrease in the drop percentages. For example in the DSDV algorithm (Figure 3), increasing the mobility level from low to medium in the tree topology resulted in drop percentages of 52% instead of 77%.

Another interesting observation is that the initial network topology can change the comparative findings. For example using an initial network topology with a bridge edge in the low mobility scenario resulted in 33% packet drop in the AODV algorithm which thus outperformed the DSDV algorithm which had a 51% packet drop. But using an initial network topology of a tree (in the low mobility scenario) reversed these results. The DSDV algorithm had a 77% packet drop, and thus outperformed the AODV algorithm which had a 86% packet drop.

V. CONCLUSION

This article explored the impact of initial network topology on the performance evaluation of four well known routing protocols for mobile ad-hoc networks. The results show that the initial network topology has a substantial impact on performance. If the initial network has a connectivity obstacle, performance is much worse than when the initial network is well connected. Algorithms that outperform other algorithms with well connected initial network topologies might not do so in different initial network topologies with connectivity obstacles. Increasing the user mobility level usually results in a decline in performance of routing algorithms in simulations that start with a well connected network topology. However when the simulation starts with a weakly connected network topology, increasing in the user mobility level enhances performance. All these findings strongly suggest that the initial network topology is an important factor in an objective simulation environment. Researchers should be aware of its influence to avoid biased results.

ACKNOWLEDGMENT

We thank Gabriel Scalosub for useful discussions.

REFERENCES

[1] The NS-2 network simulator - ns2 Website: http://www.isi.edu/nsnam/ns/

[2] M., Abolhasan, B., Hagelstein, and J. C.-P. Wang,, "Real-world performance of current proactive multi-hop mesh protocols" 2009. http://ro.uow.edu.au/infopapers/736

[3] D., Johnson and D., Maltz, "Dynamic source routing in ad hoc wireless networks, 1996,

[4] C., Perkins, E., Belding-Royer, and S., Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," IETF Mobile Ad Hoc Networking Working Group, 2007

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

[6] M., Marina, and S., Das, “On-demand Multipath Distance Vector Routing for Ad Hoc Networks,” Proc. of 9th IEEE Int. Conf. On Network Protocols, pp.14-23 (2001)

[7] D., Murray M., Dixon and T., Koziniec, "An Experimental Comparison of Routing Protocols in Multi Hop Ad Hoc Networks",

[8] S., Shah, A., Khandre, M. Shirole and G. Bhole, "Performance Evaluation of Ad Hoc Routing Protocols Using NS2 Simulation", In Mobile and Pervasive Computing (CoMPC), 2008.

[9] V., Timcenko, M., Stojanovic, and S., B., Rakas, "MANET Routing Protocols vs. Mobility Models:Performance Analysis and Comparison", In proceedings of the 9th WSEAS International Conference on APPLIED INFORMATICS AND COMMUNICATIONS (AIC '09), 2009.

[10] P., Johansson, T., Larsson and N., Hedman, "Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks", In Mobicom ‘99, Washington USA, 1999.

[11] S., Baraković, S., Kasapović, and J., Baraković, "Comparison of MANET Routing Protocols in Different Traffic and Mobility Models", In Telfor Journal, Vol. 2, No. 1, 2010.

[12] CANU Mobility Simulator Environment Website: http://canu.informatik.uni-stuttgart.de/mobisim/

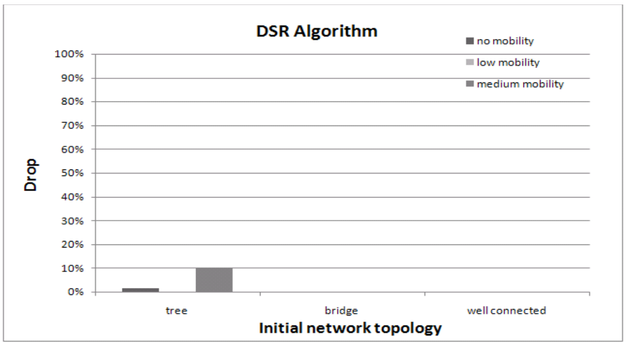

Fig. 2: DSR drops with different initial network topologies

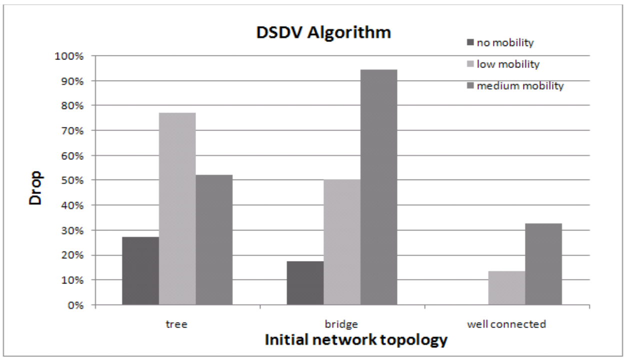

Fig. 3: DSDV drops with different initial network topologies

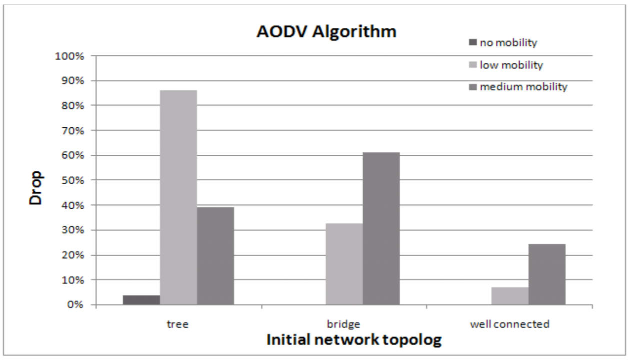

Fig. 4: AODV drops with different initial network topologies

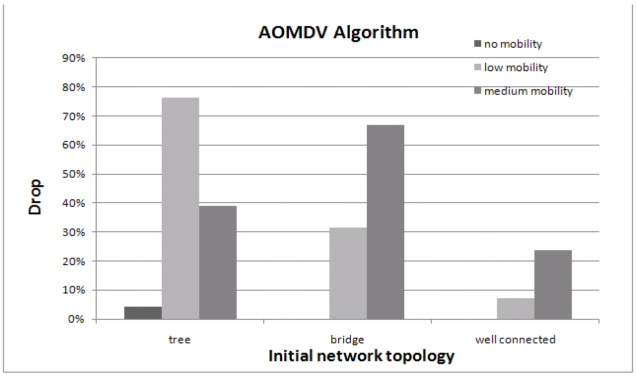

Fig. 5: AOMDV drops with different initial network topologies