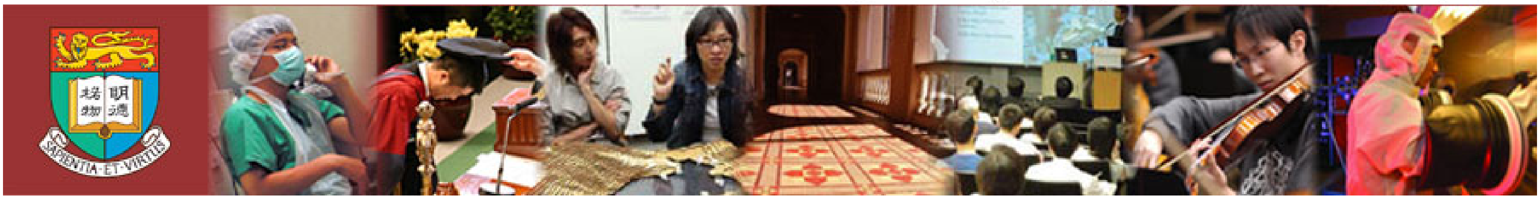

A Receiver-Initiated Soft-State Probabilistic Multicasting Protocol in Wireless Ad Hoc Networks

Lin Zhang[1,2], Dongxu Shen[2], Xiuming Shan[1], Victor O. K. Li[2], Yong Ren[1]

[1]Department of Electronic Engineering
Tsinghua University, Beijing, PRC
{linzhang, xmshan, reny}@tsinghua.edu.cn

[2]Department of Electrical and Electronic Engineering
University of Hong Kong, Pokfulam Road, Hong Kong, PRC
{dxshen, vli}@eee.hku.hk

***Abstract*— A novel Receiver-Initiated Soft-State Probabilistic multicasting protocol (RISP) for mobile ad hoc network is proposed in this paper. RISP introduces probabilistic forwarding and soft-state for making relay decisions. Multicast members periodically initiate control packets, through which intermediate nodes adjust the forwarding probability. With a probability decay function (soft-state), routes traversed by more control packets are reinforced, while the less utilized paths are gradually relinquished. In this way, RISP can adapt to node mobility: at low mobility, RISP performs similar to a tree-based protocol; at high mobility, it produces a multicast mesh in the network. Simulation results show RISP has lower delivery redundancy than mesh-based protocols, while achieving higher delivery ratio. Further, the control overhead is lower than other compared protocols.**

***Index Terms*— ad hoc networks, multicast, probabilistic forwarding**

I. Introduction and Related Work

A. MANET and Multicast

A mobile ad hoc network (MANET) is a dynamic wireless network without infrastructures, and can be flexibly deployed for scenarios when networks with infrastructure are unavailable or inadequate, such as disaster recovery, law enforcement, search and rescue, and military applications. MANETs are often organized into multiple hops, with intermediate nodes relaying traffic to destinations as coordinated by the routing protocols. In this way, nodes in MANETs are not only serving as hosts, but also as routers. It has been shown [1] that the multi-hop organization can increase network capacity and reduce transmission power. Therefore, routing is central to the functioning of MANETs. However, due to the dynamic network topology and limited resources of energy and bandwidth, routing is especially challenging for MANETs.

Much of the previous efforts on MANET routing focuses on unicasting protocols. In recent years, with the growing popularity of “community-centric” applications [2] that require the distribution of the same data to multiple recipients, such as multi-player online games and group-oriented on-demand multimedia delivery, there is an urgent need to develop multicast protocols for MANETs [3]. Although multicast has long been an important research topic for wireline networks, it is only in recent years that multicast issues are studied in MANETs. In the wireless environment, since packet delivery is prone to transmission errors and the MANET topology is dynamic, multicast faces more challenges than in wireline networks. Overall, an ideal MANET multicast protocol should minimize the signaling overhead while ensuring successful message delivery.

B. Related Work

For MANET multicasting, a straightforward approach is flooding [4]. Flooding introduces no control overhead and ensures packet delivery regardless of the mobility, but it consumes too much network resource. On the other hand, the tree-based protocols [5], [6] are direct adaptations from wireline protocols, in which a multicast tree needs to be constructed and maintained. In [5], the AMRoute (Ad hoc Multicast Routing) protocol builds a shared-tree and relies on the underlying unicast protocol to maintain connectivity among group members. It suffers from temporary routing loops and non-optimal trees with dynamic topology. In [6], the AMRIS (Ad hoc Multicasting Routing protocol utilizing Increasing id-numberS) protocol assigns an id-number for each node in a multicast session, and introduces periodical beacons to detect connection changes. However, the tree-based protocols only perform well when nodes are relatively stationary; with high node mobility, since the changing topology demands frequent reconstructions of the tree, they suffer from excessive signaling overhead and packet drops. From the perspectives of efficiency (in terms of overhead) and reliability (for packet delivery), the tree-based protocol is efficient and reliable under very low node mobility but inefficient and unreliable under high mobility, while the flooding approach is always reliable but inefficient.

The mesh-based methods [7], [8] are proposed to improve reliability by utilizing extra connections beyond the multicast tree under varying topologies. In ODMRP (On Demand Multicasting Routing Protocol) [7], the source node periodically

This research is supported in part by the Research Grant Council of Hong Kong, Grant No. HKU7044/02E

0-7803-8938-7/05/$20.00 (C) 2005 IEEE

floods control packets and receivers respond with acknowledgments upstream for mesh formation. CAMP (Core-Assisted Mesh Protocol) [8] relies on the underlying proactive routing protocol for the construction of multicast mesh. Although reliability is improved with mesh-based approaches, control packet flooding incurs high overhead. Moreover, since flooding for mesh update is costly, it shall only be performed over relatively long intervals. Thus the mesh may be outdated between successive floodings, which inevitably degrades performance, especially with high mobility. On the other hand, the mesh structure is redundant when the node mobility is low. For the extreme case of stationary nodes, the mesh structure is clearly unnecessary, and a tree structure is optimal. Obviously, the mesh-based protocols are not adaptive to node mobility.

C. *Our Contribution*

In this paper, we propose a Receiver-Initiated Soft-State Probabilistic multicast protocol (RISP) that achieves good efficiency and low overhead under both low and high mobility.

In our scheme, each intermediate node maintains a soft-state forwarding probability for the next hops. Upon receiving a multicast packet, the node probabilistically relays the packet to the next hop. This probability is continuously updated based on the periodically received control packets that are initiated by each member node in the multicast group. By the probabilistic forwarding, good reliability is achieved since redundant routes are exploited and the scheme is essentially mesh-based. Different from conventional mesh protocols such as ODMRP with a fixed mesh, our scheme generates a dynamic mesh. When node mobility is low, the mesh structure will gradually converge to a tree, thus reducing redundancy. On the other hand, our scheme employs receiver-initiated control packets, and incurs a much lower signaling overhead than source-initiated control flooding. With computer simulations, we compare the performance of our protocol with that of flooding and ODMRP. For various node mobility and multicast group sizes, our protocol has a delivery rate that is close to that of flooding and always better than that of ODMRP, and it also has the best delivery efficiency.

The rest of the paper is organized as follows. In Section II, we describe the RISP protocol. Simulation results are provided in Section III. The conclusion is given in Section IV.

II. Description of RISP

In this section, we describe the RISP protocol. Our scheme has two novel features: probabilistic forwarding and soft state.

A. *Scheme Overview: Probabilistic Forwarding and Soft Sate*

In MANET multicast protocols, each node should have knowledge on the next hop nodes for packet forwarding, and such nodes form a forwarding group. For tree-based protocols, there is only one node in the forwarding group for a destination, whereas for mesh-based protocols, there may be more than one. In conventional multicast protocols, a node will always forward the multicast packet to the nodes in the forwarding group. In other words, forwarding is performed with a probability of 1.

In our scheme, we assign each node in the forwarding group a forwarding probability. For a multicast packet, the relaying node will forward it to the next hop nodes with the designated forwarding probability. In this paper, such a forwarding strategy is called probabilistic forwarding. In contrast, the conventional forwarding policy can be viewed as deterministic forwarding, i.e., the forwarding probability is either 1 or 0.

The forwarding probability is calculated based on the periodically generated Join_Request packets from members of the multicast group. The probability computation is related to the TTL (time to live) of the Join_Request packets received from multiple neighbors, corresponding to different alternative paths to the destination. The TTL value in a Join_Request can be viewed as an indication of the hops traversed by the packet: the higher the TTL, the fewer hops traversed. When comparing Join_Requests from different neighbors, the neighbor corresponding to the path with the smallest number of hops, or the highest TTL, is preferred, since the path should be closer to the destination. Therefore, a node will compare different TTL values and update the forwarding probability accordingly.

The forwarding probability at each node is maintained in a "soft" manner, i.e., in soft-state, which means the probability is constantly refreshed. After receiving a Join_Request from a neighbor, the forwarding probability to that node is increased, since the reception of the Join_Request indicates the path through that neighbor is likely to be viable. If no Join_Request is received from a neighboring node for a certain period, the corresponding forwarding probability should be reduced. In this way, the stale routes are quickly discarded.

The above forwarding scheme can be viewed as a generalization of all conventional multicast schemes. For the extreme case that all neighbors are in the forwarding group with forwarding probability 1, the scheme is no other than flooding; when there is only one node in the forwarding group with forwarding probability 1 corresponding to one destination, it is a tree-based protocol; when there are multiples nodes each with forwarding probability 1 corresponding to one destination, the scheme is mesh-based. Obviously, RISP allows greater flexibility by having a forwarding probability between 0 and 1.

The RISP protocol ensures great adaptiveness in multicast routing. It can be expected that when the node mobility is extremely high, the protocol will degenerate into flooding; when the nodes are moving at moderate speed, the scheme produces a dynamic mesh; when nodes are almost stationary, the mesh will finally be trimmed down to a tree. This property of RISP ensures a high efficiency under all mobility scenarios.

B. *Basic Procedures*

In the RISP protocol, there are three basic procedures: multicast group setup, forwarding table maintenance, and data forwarding.

1) Multicast Group Setup: Basically, a multicast session is initialized by one source node, and there should be at least one node that is willing to join the multicast group.

After the commencement of a multicast session, the source node periodically broadcasts Beacon packets to all nodes to announce that there is a multicast session in process. In this way, all new comers can be aware of the ongoing multicast sessions. Although Beacon packets are flooded in the network, the resource consumption is comparatively low, since a Beacon packet has a small packet size. To further reduce resource consumption, Beacons are issued at relatively long intervals.

When a node decides to join a multicast session, it sends a Join_Request packet to the source node. After receiving the first Join_Request packet, the source node starts data packet transmission. The member nodes also periodically initiates Join_Request packets to the source, in order to keep the multicast session alive. If the source node has not received any Join_Request packets for a certain period, the packet transfer is terminated, since all member nodes may have quit the session. The Join_Request packet will also be used to update the forwarding probability table at each intermediate node, to be described later.

The headers of Beacon, Join_Request, and DATA packets contain the following common fields:

- Multicast Session Number (MSN): identifies the multicast session.
- Node Identity (NI): represents the identity of a sender node.
- Serial Number (SN): the serial number for a packet generated by a sender node.
- Time To Live (TTL): indicates the number of hops the packet could travel before timeout.

The combination of NI and SN uniquely identifies a packet, and are used to prevent the formation of routing loops. When a packet has been transferred by a node, the tuple (NI, SN) is stored in a cache. For each incoming packet, (NI, SN) is first compared with the records in the cache. If there is a match, the packet must have looped, and is thus discarded.

2) Forwarding Table Maintenance: At each node, there are two reliability tables to be maintained for each multicast session. One is a downstream table for the delivery of DATA packets, and the other is an upstream table for the transfer of Join_Request packets. The maintenance of the two tables are essentially similar, and we focus on the illustration of the downstream table maintenance.

The downstream table stores the probability that characterizes the reliability of a route through a neighbor node. The reliability is determined from Join_Request packets delivered from that neighbor. If a neighbor continuously relays Join_Request packets, it is highly likely a member node can be reached via that neighbor. On the contrary, if there is no Join_Request packet from a neighbor for a long period, it is a strong indication that there is no route through that neighbor. Therefore, the probability should be a monotonically increasing function of the frequency of Join_Request packets.

To compute the route reliable probability, we devise two functions. The first is the probability refresh function from a neighbor upon the reception of a Join_Request packet, or in short, Refresh Function. The second one is the probability decay function with respect to time (Decay Function).

Obviously, when a Join_Request packet is received from a neighbor, the route reliability for that neighbor should be increased. The exact amount of increment is determined by the TTL value in the packet header. When multiple Join_Request packets are received from different neighbors, a larger TTL indicates the route through that neighbor is closer to the destination node. Therefore, the Refresh Function should be a monotonically increasing function of TTL. In this paper, the route reliability probability is given by the Refresh Function in the form of

$$p_r = 1 - e^{-\alpha \cdot \lambda}$$

where α is a constant coefficient called probability growth rate, and λ is a value determined by the TTL value T. Whenever a new Join_Request arrives, λ is updated as

$$\lambda' = \lambda + \Delta$$

where $\Delta = d + \frac{D-d}{T_{max}} \cdot T$ is called the *reinforcement strength*.

The route reliability probability also decays with respect to time. The Decay Function is defined as

$$\lambda' = \lambda - \beta \cdot t$$

where β is the decay rate, and t is the time elapsed since the last arrival of Join_Request. Note that λ is set to zero once it decays to a negative value. The composite effect of the Refresh Function and Decay Function is to foster reliable paths and weed out unreliable ones. When a node sees a DATA or Join_Request packet of a multicast session for the first time, all entries of the forward probability table is initialized to 1. With the elapse of time, paths with frequent Join_Request packet arrivals are reinforced to stay at a high reliability probability. Due to the decay function, paths that do not have Join_Request updates gradually decline to low probabilities. Therefore, influenced by the Refresh Function and Decay Function, RISP will converge to tree-based forwarding with a stationary topology.

The node's arbitrary movement results in frequent topology changes, which require timely updates to implement proper multicast routing. In RISP, when connections fail or new connections appear, adjustment is made according to the following two rules.

- **Local flooding rule:** When the MAC layer reports a connection failure, RISP checks the forwarding probability of such a link. If it is higher than a theshold p_h(to be explained in the next subsection), the probability corresponding to all other links in that forwarding table are reset to 1 and the entry corresponding to the failed connection is deleted; otherwise, RISP simply deletes the failed connection. This procedure causes a local flooding when failures on valid connections are detected. This is because in MANETs with high node density, most link failures may be overcome by by-pass forwarding.
- **Search-new-connection-first rule:** When a new connection to a node is discovered, RISP appends an entry to both upstream and downstream tables and fill each entry with the value of 1. With this rule, RISP utilizes a newly discovered path immediately.

With the two rules described above, RISP adapts to path changes caused by user mobility.

3) DATA Forwarding: Based on the route reliability, the forwarding probability can be derived. We conduct a piecewise non-linear mapping of the reliability to calculate the actual forwarding probability, according to which the packets are forwarded.

Let p_f denote the forwarding probability. There is

$$p_f = \begin{cases} 1 & p_r \geq p_h \\ p_r & p_l < p_r < p_h \\ 0 & p_r \leq p_l \end{cases}$$

where p_h and p_l are two threshold values with $p_h > p_l$.

The use of such a mapping function is intuitive. When the route reliability is higher than a threshold, the link is highly reliable, and packets are always forwarded to that direction. When the reliability is lower than a threshold, it is likely that the forwarding will be a waste of resource. Therefore, there is no forwarding to that direction. When the reliability lies in between, the forwarding probability is equal to the reliability value. With such a mapping, we ensure that packets are always forwarded to the most reliable paths, and not forwarded to the unreliable paths.

III. Performance Evaluation

A. Simulation Environment and Performance Metrics

We evaluate the performance of RISP through simulation in a variety of mobility and multicast communication scenarios using NS-2 network simulator[12] with mobile NS-2 extensions [13]. Since the performance of ODMRP has been thoroughly studied and compared to other tree-based and mesh-based ad hoc multicasting protocols in [14], we use it as a reference for evaluating RISP. Flooding is also simulated since it has the best robustness and worst efficiency.

We simulate a MANET with 100 nodes in a 1000m x 1000m area, and each run lasts 1000 seconds. The movement model of the nodes in our simulations is the Random Waypoint model. Each node independently starts at a random location in the simulation area and remains stationary for a period of pause time. The node then generates a uniformly distributed new location to move to. At the same time, the node randomly selects a speed value from a speed range[1], 1-20 m/s in our paper, and move to the new location at the selected speed. This movement pattern is repeated for the duration of the simulation.

The source node generates CBR traffic. We also assume that all members of the multicast group join the group at the start of the simulation, and stay in the multicast session for the duration of the simulation. Thus we do not consider node departures.

The above movement scenario, together with the multicast traffic scenario, are generated once for each run, and are adopted by all three protocols for a fair comparison.

The following metrics are used for the evaluation of RISP, ODMRP and flooding :

- **Packet delivery ratio**: The average number of multicast DATA packets actually received by one destination node over the number of transmitted packets from the source. For example, with five destinations and 1000 generated DATA packets at the source, if 4500 packets[2] are received in total by the 5 destinations, or 900 on average for each node, the delivery ratio is 0.9.
- **Delivery redundancy**: The average number of DATA packet relays per received DATA packet. This metric represents the efficiency of the multicast forwarding.
- **Control overhead**: The average number of control packet relays per received DATA packet.

B. Performance under Different Movement Scenarios

We study the performance of RISP, ODMRP, and Flooding under different movement scenarios. With a speed range of 1-20m/s, movement scenarios are generated for 8 different pause times: 1, 30, 60, 120, 300, 600, 900 and 1000 seconds. Note that a pause time of 900 and 1000 seconds represent almost static topologies, whereas nodes pausing at 1 second are basically constantly moving. Ten scenarios are generated for each pause time to produce an average. The multicast group has 1 source and 10 destinations.

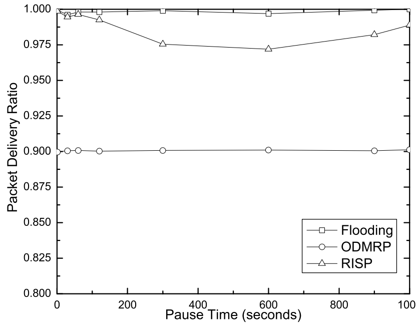

Fig. 1. Delivery ratio under different movement scenarios

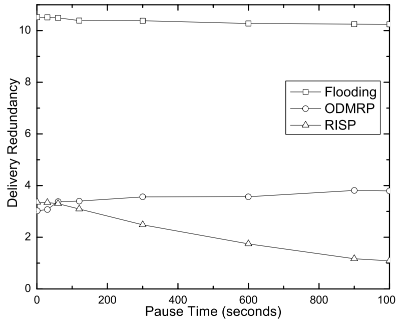

Fig. 2. Delivery redundancy under different movement scenarios

The simulation results in Fig. 1 show that under all movement scenarios, RISP has a consistently higher delivery ratio than ODMRP, approaching that of Flooding. Fig. 2 reveals

[1] To avoid the average speed decay problem [15], we set a minimum speed for the nodes.

[2] Note multiple receptions of the same DATA packet is counted once.

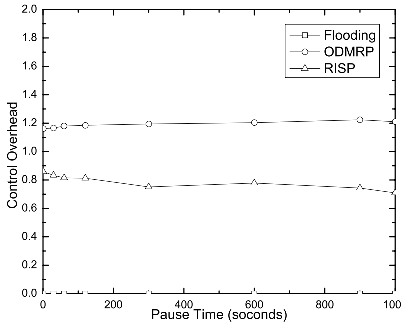

Fig. 3. Control overheads under different movement scenarios

that when the node mobility is low, the delivery redundancy of RISP is much lower than (as low as 25%) that of ODMRP. This is natural, since under low mobility, RISP functions similarly to a tree-based protocol, and is thus more efficient than the mesh produced by ODMRP. When node mobility is extremely high, the redundancy is almost the same (slighly higher) as that of ODMRP. This is because RISP evolves to a mesh-based protocol with increased mobility. As expected, Flooding has the worst delivery redundancy under all circumstances.

As shown in Fig. 3, RISP has much lower control overhead than ODMRP. This is because in RISP only multicast members issue the Join_Request control packets, while ODMRP employs flooding from the source to refresh the routing information.

C. Performance under Different Multicast Group Size

We study the influence of multicast group size on delivery redundancy. In Fig. 4, the delivery redundancy is plotted against the *multicast group member density*, which is the multicast group size normalized by the total number of nodes. In the simulation, the total number of nodes is 50, and the nodes are stationary.

For all three protocols, the delivery redundancy declines with higher member density. With more nodes belonging to the multicast group, one transmission can reach more destinations, thus reducing redundancy. Not surpisingly, RISP still has the least delivery redundancy.

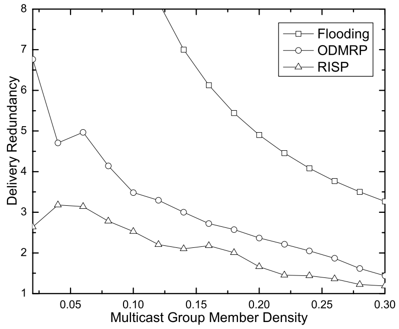

Fig. 4. Delivery redundancy with different multicast group density.

IV. Conclusions

A novel Receiver-Initiated Soft-State Probabilistic multicasting protocol (RISP) for MANET is proposed in this paper. RISP introduces forwarding probability and soft-state in making the forwarding decision. Aided by the receiver initiated control packets, it is adaptive to node mobility, and can always ensure a high delivery ratio. At low speed, its efficiency is much higher than that of mesh-based protocols such as ODMRP, since it performs like a tree-based protocol. At high speed, the delivery ratio is guaranteed because it functions as a mesh-based protocol. The control overhead is also low because only multicast members initiate control packets. The above claims have been confirmed by simulations. In the future, we plan to further improve the performance of RISP by tuning the parameters of the protocol under different movement and traffic scenarios.

References

[1] S. Toumpis and A. J. Goldsmith, “Capacity Region for Wireless Ad Hoc Networks,” *IEEE Transactions on Wireless Communications*, vol. 2, no. 4, July 2003, pp. 736-748.

[2] C. de Morias Cordeiro, H. Gossain, and D. P. Agrawal, “Multicast over Wireless Mobile Ad Hoc Networks: Present and Future Directions,” *IEEE Network*, vol. 17, issue. 1, Jan 2003, pp. 52-59.

[3] P. Mohapatra, C. Gui, and J Li, “Group Communication in Mobile Ad Hoc Networks,” *IEEE Computer Magazine*, vol. 37, issue. 2, Feb. 2004, pp. 52-59.

[4] K. Obraczka, K. Viswanath, and G. Tsudik, “Flooding for Reliable Multicast in Multi-hop Ad Hoc networks,” *Wireless Networks*, vol. 7, issue. 6, Nov. 2001, pp. 627-634.

[5] E. Bommaiah, M. Liu, A. McAuley, and R. Talpade, “AMRoute: Adhoc Multicast Routing Protocol,” *Internet draft*, IETF, Aug. 1998.

[6] C. W. Wu, Y. C. Tay, and C. K. Toh, “Ad hoc Multicast Routing protocol utilizing Increasing id-numberS (AMRIS) Functional Specification,” *Internet draft*, Nov. 1998.

[7] S. J. Lee, M. Gerla, and C. C. Chiang, “On-Demand Multicast Routing Protocol,” In *Proceedings of IEEE WCNC’99*, New Orleans, LA, Sep. 1999, pp. 1298-1304.

[8] J. J. Garcia-Luna-Aceves and E. L. Madruga, “The Core-Assisted Mesh Protocol,” *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, Aug. 1999, pp. 1380-1394.

[9] M. Mauve, H. FüSSler, J. Widmer, and T. Lang, “Position-Based Multicast Routing for Mobile Ad Hoc Networks,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 7, issue. 3, July 2003, pp. 53-55.

[10] R. Chandra, V. Ramasubramanian, and K. Birman, “Anonymous Gossip: Improving Multicast Reliability in Mobile A d Hoc Networks,” In *Proceedings of IEEE ICDCS-21*, Phoenix, AZ, April 2001, pp. 275-283.

[11] J. Luo, P. T. Eugster, and J. P. Hubaux, “Route Driven Gossip: Probabilistic Reliable Multicast in Ad Hoc Networks,” In *Proceedings of IEEE Infocom 2003*, San Francisco, CA, Mar. 2003, pp. 2229-2239.

[12] The VINT project, The Network Simulator ns-2, *http://www.isi.edu/nsnam/ns/*

[13] The Rice Monarch Project. The Rice Monarch extensions to the ns simulator, *http://www.monarch.cs.rice.edu/cmu-ns.html*

[14] S. J. Lee, W. Su, J. Hsu, M. Gerla, and R. Bagrodia, “A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols,” In *Proceedings of IEEE Infocom 2000*, Tel Aviv, Israel, March 2000, pp. 565-574.

[15] J. Yoon, M Liu, and B Noble, “Random waypoint considered harmful,” In *Proceedings of IEEE Infocom 2003*, San Francisco, CA, March 2003, pp. 1312-1321.