An Ant-based Multicasting Protocol in Mobile Ad Hoc Networks*

Lin Zhang[1,2], Dongxu Shen[2], Xiuming Shan[1], Victor O. K. Li[2]
[1]Department of Electronic Engineering
Tsinghua University, Beijing, PRC
{linzhang, xmshan}@tsinghua.edu.cn
[2]Department of Electrical and Electronic Engineering
University of Hong Kong, Pokfulam Road, Hong Kong, PRC
{dxshen, vli}@eee.hku.hk

Multicasting protocols deliver data packets from a source node to multiple receivers, and serve a very important function in mobile ad hoc networks (MANETs). In this paper, a novel Receiver-Initiated Soft-State Probabilistic multicasting protocol (RISP) for MANETs is proposed. RISP is inspired by the ant colony's route seeking mechanism, in which an individual ant chooses the optimal path to its destination through cooperation with others in a totally distributed manner. Imitating the behavior of ants in nature, RISP introduces probabilistic forwarding and soft-state for making relay decisions that is automatically adaptive to node mobility in MANETs. Compared with other protocols, we show by computer simulations that RISP has lower delivery redundancy, while achieving higher delivery ratio at all mobility scenarios. Furthermore, RISP has lower control overhead.

Keywords: ad hoc networks, multicast, ant colony optimization

1. Introduction and Related Work

A mobile ad hoc network is a dynamic wireless network without a stationary infrastructure, and can be flexibly deployed for scenarios when networks with stationary infrastructure are unavailable or inadequate, such as disaster recovery, law enforcement, search and rescue, and military applications. MANETs are often organized with multiple hops, with intermediate nodes relaying traffic to receivers 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.

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 multicasting protocols for MANETs [3]. Although multicast has long been an important research topic for wireline networks, it is only

*This research is supported in part by the Research Grant Council of the Hong Kong Special Administrative Region, China, under Grant No. HKU7044/02E.

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 in MANETs faces more challenges than in wireline networks. Overall, an ideal MANET multicasting protocol should minimize the signaling overhead while ensuring successful message delivery.

For MANET multicasting, a straightforward approach is to apply flooding [4]. Although flooding avoids control overhead and ensures packet delivery regardless of the mobility, 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. However, 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 (in 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. Although reliability is improved with mesh-based approaches, 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.

In this paper, we propose a Receiver-Initiated Soft-State Probabilistic multicasting protocol. The idea of RISP is inspired by the route seeking behavior of the ant colony, in which individual ants are capable of finding the best path to the food sources or the nest without global administration, i.e., in a distributed way, which is desirable for MANET protocols. RISP imitates the ant colony's behavior of route seeking, with each intermediate node maintaining soft-state forwarding probability tables for the next hops. Upon receiving a multicast packet, the node relays it probabilistically with the coordination of probability tables, which are continuously updated according to periodically received control packets initiated by each member node in the multicast group. By probabilistic forwarding, good reliability is achieved since redundant routes are exploited and the scheme is essentially mesh-based. In contrast to conventional mesh-based protocols such as ODMRP [7] 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. At the same time, our scheme employs receiver-initiated control packets, and incurs a much lower signaling overhead than source-initiated control flooding. In 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 introduce the route seeking behavior of ant colonies to explain the rationale of RISP. In Section III, the protocol is presented in detail. Simulation results are provided in Section IV. Section V concludes the paper.

2. The Ant-based Routing Algorithm

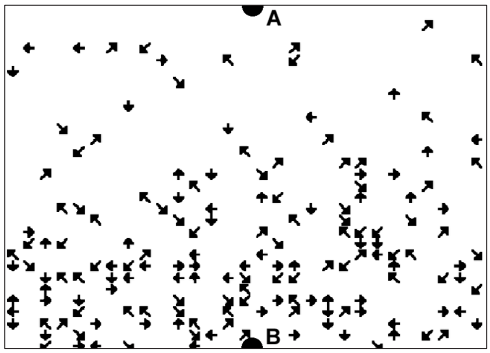
The ant, as a social insect, is capable of seeking routes between the nest and multiple food sources. They accomplish the mission with great efficiency and, in case of environmental changes, can quickly discover new routes. Interestingly, biologic research shows that the ant colony achieves this without central administration. Each individual ant in the colony obeys some simple rules to decide the direction to travel, based on the information collected from the neighborhood. The rules are:

(1) Upon leaving the nest or a food source, the ant begins to deposit to the surroundings a chemical substance called *pheromone.* The quantity of the pheromone deposited is decreased as the ant travels further and further away.
(2) An ant chooses its next move with a preference to the direction with a higher quantity of pheromone corresponding to its target, which is either the nest or a food source, depending on whether it is carrying food.
(3) If the move is not feasible, it randomly chooses another feasible direction to move on.
(4) The quantity of pheromone in the surroundings is reduced at a constant rate, as a consequence of evaporation.

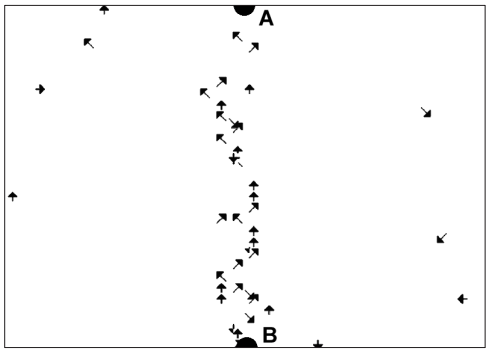
With Rule 1, an ant colony will generate a field of pheromone in the surroundings, such that, at any point, the direction with the shortest distance to the destination has the highest density of pheromone corresponding to that destination. As a result of Rule 2, ants will travel along the shortest path to their destinations. Rules 3 and 4 are to deal with route changes. When a route fails, Rule 3 will conduct a local flooding to find an alternate route and Rule 4 guarantees a bad route will be eliminated quickly.

At the very beginning, due to the absence of pheromone distribution, the ant colony floods its ants all around the neighborhood to search for the food source. Once some ants reach the food source and go back to the nest, the shortest path will be established. A 2-D cellular automata was introduced in [9] to simulate the ant colony routing algorithm for wireline IP unicast networks. Fig.1 shows the distribution of ants at 0.5 second and 5 second (each arrow represents an ant, with the arrow pointing to the direction of travel), and we can see that a straight route emerges between the nest B and the food source A.

In MANETs, since a global and centralized administration is undesirable due to its vulnerability to single-point failure, distributed protocols are mandatory. Ant-based routing, which is distributed and, thus, scalable, motivates routing design in MANETs. It has been studied in wireline networks for unicast routing [9,10,11,12].



(a) t=0.5



(b) t=5

Figure 1. The Route Convergence of An Ant Colony

However, we are unaware of the implementation of the ant-based routing algorithm in MANETs, especially for multicast. We will introduce such a protocol in this paper.

3. Description of RISP

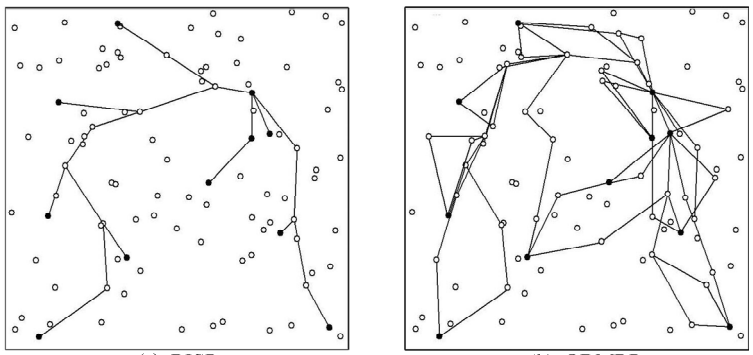
In this section, we describe the RISP protocol. Our scheme has two novel features: probabilistic forwarding and soft state.

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

First of all, we make an analogy between the multicast routing of MANETs and the routing mechanism of the ant colony. We consider the source node of the multicast group as the nest, and the receivers as the food sources. Control packets sent by the source node or the receivers may be considered as the ants travelling from the nest or the food sources. The Time To Live (TTL) value of the packet is a metric

of the "closeness" to the destination, or the distance an ant has travelled from the nest or the food resources.

In RISP, we assign each node a forwarding probability table corresponding to each receiver in the multicast session. For a multicast packet, the relaying node will forward it to the next hop nodes with the designated forwarding probability. In the implementation of the protocol, upon receiving a packet, a relaying node generates a random number uniformly distributed between 0 and 1 for each neighboring node. If this number is smaller than the forwarding probability, the corresponding neighboring node is then chosen as a next hop node of this packet. The list of the next hop nodes, embedded in the RISP frame header, is broadcasted with the packet. If no collision occurs, all the neighbor nodes will receive the packet but only the nodes in the next hop list will relay the packet. In this paper, such a forwarding strategy is called *probabilistic forwarding*. It is clear that such a strategy is not collision-free, but the overall performance can still be guaranteed by the redundancy of probabilistic forwarding. In contrast, the conventional forwarding policy can be viewed as deterministic forwarding, i.e., the forwarding probability is either 1 or 0.



(a) RISP (b) ODMRP

Figure 2. Forward Group Configuration
(With stationary topology; the number of nodes is 100; the number of multicast group members, shown in solid dots, is 11, with one source node and 10 receivers.)

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 of the Join_Request packets received from multiple neighbors, corresponding to different alternative paths to the receiver. By 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 receiver. Therefore, a node will 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 while all the others are decreased, 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 RISP protocol ensures great adaptiveness in multicast routing. When the node mobility is extremely high, the protocol will degenerate into flooding; when the nodes have moderate mobility, 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 high efficiency under all mobility scenarios.

Since the performance of ODMRP, a mesh-based multicasting protocol, has been thoroughly studied and enjoys the best performance compared to other tree-based and mesh-based ad hoc multicasting protocols in [13], we use it as a reference for evaluating RISP. Fig. 2 shows the comparison between the multicast group configuration of RISP and ODMRP, a mesh-based multicast protocol, when the nodes are stationary. It is clear that RISP has higher efficiency.

3.2. *Basic Procedures*

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

3.2.1. *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 progress. In this way, all newcomers 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 as 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. The SN is 16 bits, allowing 65,536 unique numbers before a repetition.
- Time To Live (TTL): indicates the number of hops the packet could travel before being dropped.

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.

3.2.2. *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 and Beacon 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 downstream table maintenance.

The downstream tables store 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 that 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 as shown in Fig. 3. 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 growth 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 receiver. 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

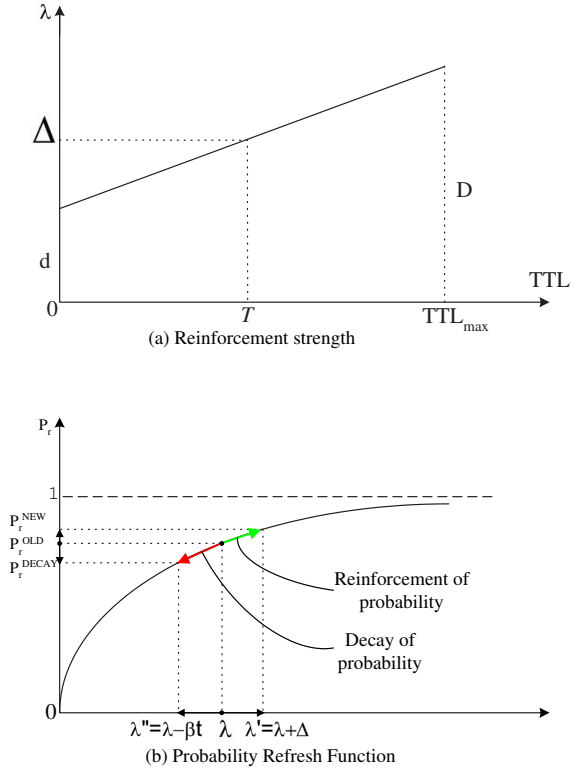


Figure 3. Reinforcement strength and probability refresh function

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 Beacon 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 and will stay at a high reliability probability. Due to the decay function, paths that do not have Join_Request updates gradually de-

cline 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 mobility results in topology changes, which require timely updates in 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 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 can quickly adapt to path changes caused by user mobility.

3.2.3. *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.

4. Performance Evaluation

4.1. *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 [13] with mobile NS-2 extensions [14]. We use ODMRP 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 runs lasts 1000 seconds. The maximum transmission range of each node is 250 meters. 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[a], 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 (Constant Bit Rate) traffic. The packet size is 1024 bytes and the packet rate is 4 packets/s. 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.

There are 10 CBR streams running simultaneously in the network as background traffic. Each background CBR stream is assigned to a pair of randomly selected source and destination nodes, with the condition that all these 20 nodes are different. The packet size of the background CBR stream is 512 bytes and the packet rate is 4 packets/s.

The Beacon and Join_Request packets are sent periodically with a period of 3 seconds. We choose this value to do a comparative study with ODMRP, whose control packets are sent with the same period.

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 receiver node as a fraction of the number of transmitted packets from the source. For example, with five receivers and 1000 generated DATA packets at the source, if 4500 packets[b] are received in total by the 5 receivers, or 900 on average for each node, the delivery ratio is 0.9.
- **Delivery redundancy**: The average number of DATA packet relays per re-

[a]To avoid the average speed decay problem [15], we set a minimum speed for the nodes. The pause time and speed are uniformly distributed.
[b]Note multiple receptions of the same DATA packet is counted once.

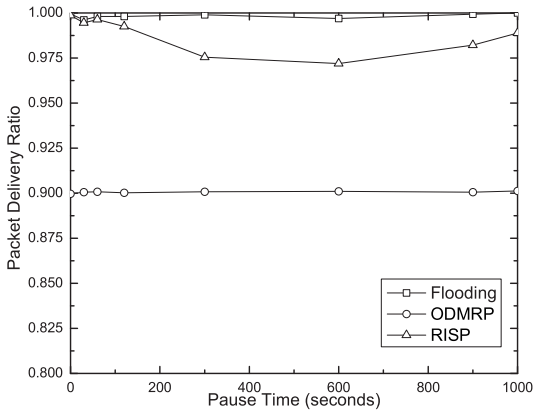


Figure 4. Delivery ratio under different movement scenarios

ceived DATA packet per hop. This metric represents the efficiency of the multicast forwarding scheme. For example, with one receiver and 1000 generated DATA packets at the source in a stationary network, if the minimum number of hops between the source node and the receiver node is 4, and if we encounter 4500 DATA packets sending or forwarding events in the network and 1000 DATA packet receiving events by the receiver, the delivery redundancy is $\frac{4500}{1000\times 4} = 1.125$. The reason we normalized with the number of minimum hops in the metric is to make it independent of the network topology.

- **Control overhead**: The average number of control packet relays per received DATA packet.
- **Delay:** The average delay of all the data packets received.

4.2. *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 mean pause times: 0, 30, 60, 120, 300, 600, 900 and 1000 seconds. Note that a pause time of 1000 seconds represents a static topology, whereas nodes with a pause time of 0 seconds are constantly moving. Ten scenarios are generated for each pause time to produce an average. The multicast group has 1 source and 10 receivers.

The simulation results in Fig. 4 show that under all movement scenarios, RISP has a consistently higher delivery ratio than ODMRP, approaching that of Flooding. Fig. 5 reveals that when the node mobility is low, the delivery redundancy of RISP is much lower than (as low as 50%) that of ODMRP. This is natural, since under low mobility, RISP functions similarly to a tree-based protocol, and is thus more

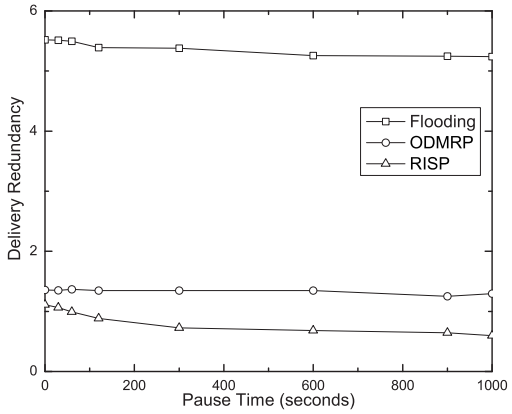


Figure 5. Delivery redundancy under different movement scenarios

efficient than the mesh produced by ODMRP. When node mobility is extremely high, the redundancy is almost the same 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.

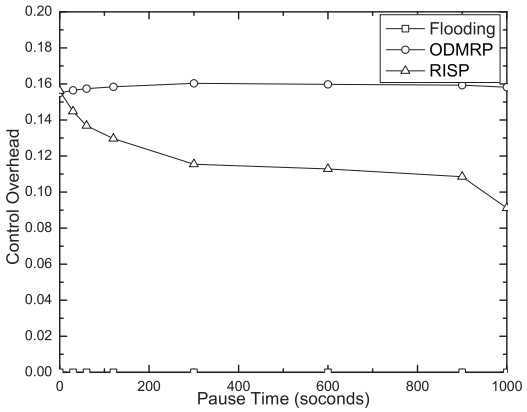


Figure 6. Control overheads under different movement scenarios

As shown in Fig. 6, RISP has much lower control overhead than ODMRP. This is because in RISP only multicast members issue the Join_Request control packets,

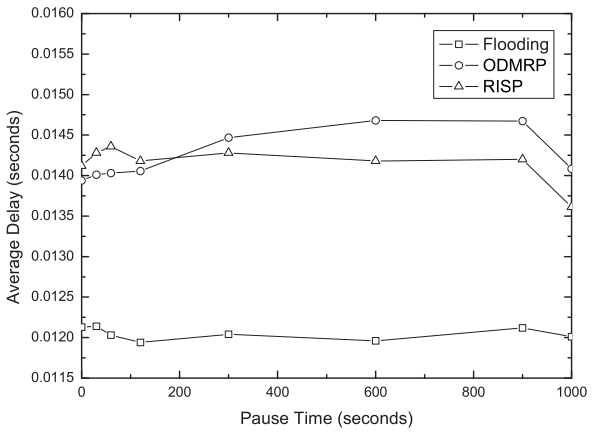


Figure 7. Delay under different movement scenarios

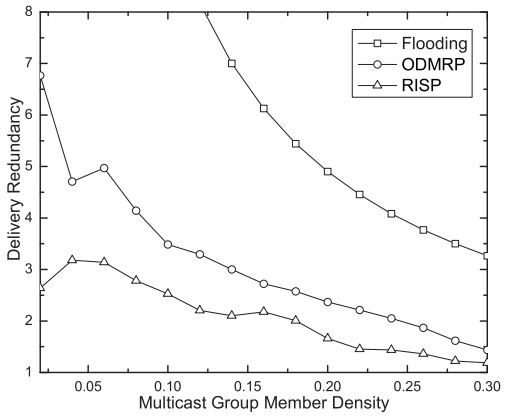


Figure 8. Delivery redundancy with different multicast group density.

while ODMRP employs flooding from the source to refresh the routing information. Fig. 7 shows the average delay of all DATA packets. It is not surprising that flooding has the lowest delay because it utilizes all the possible routes. RISP and ODMRP have similar delay characteristics.

4.3. *Performance under Different Multicast Group Size*

We study the influence of multicast group size on delivery redundancy. In Fig. 8, 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.

5. Conclusions

Inspired by the ant colony routing mechanism, a novel Receiver-Initiated Soft-State Probabilistic multicasting protocol (RISP) for MANETs is proposed in this paper. RISP introduces forwarding probability and soft-state in making the forwarding decision. Aided by 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.

Acknowledgment

We would like to thank the two anonymous reviewers for suggesting improvements and their helpful comments.

Bibliography

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. L. Zhang, Y. Ren, X.M. Shan, "Pheromone Based Artificial-life Routing System for IP Networks," *Tsinghua Science and Technology*, vol. 9, no. 2, April 2004, pp. 213-218.
10. R. Schoonderwoerd, "Ant-based load balancing in telecommunications networks," *Adaptive Behavior*, vol. 5, no. 2, 1996, pp. 169-207.
11. G. Di Caro, M. Dorigo, "AntNet: Distributed Stigmergetic Control for Communications Networks," *Journal of Artificial Intelligence Research*, vol. 9, Sep. 1998, pp. 317-365.
12. D. Subramanian, P. Druschel, and J. Chen, "Ants and Reinforcement Learning: A case study in routing in dynamic networks," *http://cs-tr.cs.rice.edu/Dienst/UI/2.0/Describe/ncstrl.rice_cs/TR96-259*, July 1998.
13. 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.The VINT project, The Network Simulator ns-2, *http://www.isi.edu/nsnam/ns/*
14. The Rice Monarch Project. The Rice Monarch extensions to the ns simulator, *http://www.monarch.cs.rice.edu/cmu-ns.html*
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.