Adaptive Routing in Dynamic Ad Hoc Networks

Cong Liu and Jie Wu
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431
{cliu8@, jie@cse.}fau.edu

***Abstract*—Dynamic ad hoc networks are mobile ad hoc networks (MANETs) where network characteristics, such as network density and node mobility, change significantly over time and space. Sometimes, dynamic ad hoc networks resemble a dense ad hoc network. At other times, they resemble a delay tolerant network. Many real networks follow the paradigm of dynamic ad hoc networks. Military networks, wildlife tracking sensor networks, and vehicle networks are some of these examples. In dynamic ad hoc networks, conventional routing schemes fail when the network characteristics do not fall into their applicable scenarios. Previous research has proposed a variety of routing schemes for each specific network scenario. For instance, distributed routing tables are built for efficient multi-hop, single-copy routing in static and dense networks. Mobility assisted, multi-copy routings are proposed in sparse networks where contemporary paths might not exist. With the advantages of the existing schemes in mind, we introduce a new routing scheme, Adaptive ROuting in Dynamic ad hoc networks (AROD), which is a seamless integration of several existing schemes. Simulation results show that AROD is highly scalable and is adaptive to different network scenarios.**

Keywords: Adaptive routing protocol, dynamic ad hoc networks, delay tolerant networks (DTNs).

I. Introduction

In mobile ad hoc networks (MANETs), two nodes can exchange data when they are located within one another's communication range. A node can deliver data to another node directly or via intermediate nodes without relying on base-stations. Traditional ad hoc routing uses a single-copy, multi-hop delivery scheme under the assumption of the existence of contemporary source-destination paths and unlimited network capacity.

Interest has grown over the past few years in delay tolerant networks (DTNs). DTNs are usually sparse, such that a contemporary path between a source and a destination might not exist, and delivery of messages must utilize node mobility. A path existing between a source and a destination in a DTN means that the source and the destination are connected in the overlapped evolving network graphs. The primary focus of existing DTN routing protocols is to increase the likelihood of finding such a path, in extremely sparse networks with extremely limited information. To this end, a variety of mechanisms are used, including estimating meeting probabilities, message replication, network coding, placement of stationary storage devices, and using prior knowledge of mobility.

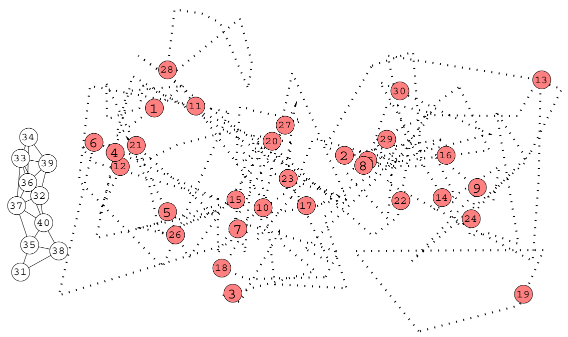


Fig. 1. A dynamic ad hoc network. Red nodes with random trajectories form a sparse sub-network, and white static nodes form a dense one.

Our interests lie in developing an algorithm that is as general as possible. We model *dynamic ad hoc networks* as MANETs which are connected most of the time but are not always disconnected due to mobility. In a dynamic ad hoc network, network characteristics can change significantly over time and space. At certain times the mobile nodes might gather, enabling the instant communication between the nodes using direct or multi-hop delivery. At other times, nodes might spread and roam around a large geographical region, and it is appropriate to deliver messages in a store-and-forward, multi-copy scheme to increase delivery probability and decrease delivery time. Many real networks follow the paradigm of dynamic ad hoc networks. Military networks [1], wildlife tracking sensor networks [8], and vehicle networks [3] [12] are some of these examples. An example of a dynamic ad hoc network is shown in Figure 1, where some nodes that are sparsely deployed with randomized trajectories form a DTN sub-network, while other nodes stay close to each other and form a dense sub-network.

Different existing routing protocols would work well for the different scenarios exhibited by a dynamic ad hoc network. However, it is inconvenient to require the users to switch between multiple routing protocols. Moreover, if different scenarios are exhibited by different parts of a network, the routing protocols used must be able to communicate and cooperate with each other, which is another difficult task. Thus, a routing protocol that is adaptive in an effor to maintain good performance and that also operates seamlessly in different network scenarios is desired. In this preliminary work, we investigate an adaptive routing algorithm in dynamic ad hoc networks, aiming to show that such a protocol is possible rather than proposing a routing protocol to optimize the routing performance in different scenarios.

The goal of our algorithm is to be adaptive to different network densities and different mobility models. We broadly discriminate two kinds of mobility models: local mobility and random mobility. In local mobility, each node has a home region which it visits more often than other regions. As a result, it meets some nodes more frequently than others. Local mobility also includes the situations where the nodes do not have preferred regions, but the dissemination of the meeting information is fast enough to form gradients of delivery probabilities among the nodes. Random mobility refers to the opposite situation where the motion of the nodes is fast and random, which leads to difficulties when attempting to estimate delivery probability in the network.

The design principle of our proposed *Adaptive ROuting in Dynamic ad hoc networks* (AROD) is exemplified as follows. In a network that has an adequate communication capacity (i.e., the total transfer opportunities in the network) and a clear gradient of decreasing estimated delivery latency or increasing delivery probability to each destination, such as a dense network with a local mobility pattern, it is suffice to use a single-copy and multi-hop delivery. However, in a sparse mobile network with random mobility and limited transfer opportunities, mobility-assisted and multi-copy delivery is used to shorten the delivery time and increase the delivery ratio. AROD adaptively trades off delivery latency/probability to bandwidth consumption. It is a seamless integration of the different routing schemes used for different network scenarios.

The remainder of this paper is presented as follows. In Section II, we go over some related work and summarize our contributions. Section III presents our proposed solution, AROD, and its implementation details. Section IV shows the adaptive performance of AROD in different network scenarios. Finally, Section V concludes the paper and discusses ideas for future work.

II. Related Works and Contributions

This paper focus on routing protocols in mobile ad hoc networks that do not rely on particular hardware support or prior knowledge, such as a GPS that provides a node with its position, a powerful global channel to disseminate the status information of the nodes, or a bounded network area. Previous work on MANETs has been based on various assumptions regarding node density and mobility models. Conventional ad hoc network routing schemes such as DSR [7], AODV [5], and DSDV [13] are proposed in dense networks where contemporary source-destination paths exist.

In delay tolerant networks [6], especially the extremely sparse networks where the average node degree is smaller than 1, messages can still be delivered if paths exist in the evolving graph of the network. Existing routing schemes such as Epidemic [15], Prophet [10], Spray and Wait [14], Spray and Focus [14], MaxProp [3], and RAPID [2], use a store-carry-forward scheme.

Previous proposed adaptive routing protocol includes CAR [11], where routing methods are selected depending on whether the recipient presents in the same connected component (cloud) in the network. If it does, the message is delivered by DSDV [13]. Otherwise, the message is sent to the node in the cloud which has the highest delivery probability. This protocol, however, uses pure single-copy forwarding and works well only for local mobility.

Routing information is exchanged by peers in the control channel of AROD. Our approach is built on several important insights from previous works. Chen and Nahrstedt [4] and Spyropoulos et al. [14] use replicas to decrease average delay and increase delivery rates. Leguay et al. [9], Burgess et al. [3], Levine et al. [2], and Leguay et al. [14] suggest using historical connectivity information and predictions of future connectivity information in order to improve routing performance. Burgess et al. [3] shows that flooding acknowledgements effectively reduce delays and increase delivery rates by freeing up resources used by delivered packages. Mirco et al. [11] uses proactive routing to send messages to destinations within the same cloud, and to predict forwarding nodes for destinations in other clouds.

Our main contribution in this work is demonstrating the feasibility of an adaptive routing approach in dynamic ad hoc networks. To this end, we:

- present a routing protocol, AROD, which is the first routing scheme that is adaptive to network density as well as to mobility patterns,
- show that AROD behaves appropriately and maintains desired properties in different network scenarios, and
- implement AROD and show its efficacy in different network scenarios.

III. Adaptive Routing in Dynamic Ad Hoc Networks

Two nodes transfer data messages to each other when they are within one another's communication range. During a transfer, the sender replicates messages while retaining a copy. Messages may not be fragmented. We assume unlimited storage capacity, and that a node never deletes messages until it receives an acknowledgement or timeout.

Each message is given a Time-To-Live (TTL) which specifies a timeout of the message after which a message is no longer meaningful and can thusly be dropped. Two nodes are in the same cloud if there is a contemporary multi-hop contemporary path.

Similar to [14], we give each message a logical floating-point ticket which is initialized 1.0. Whenever a message is delivered, both the sender and the receivers hold a complete copy of the message while the new tickets associated with their copies in the sender and the receivers add up to the ticket of the original message in the sender.

A. AROD Design

AROD's adaptation to the correct forwarding strategy is embodied by the formulation of message priority which is

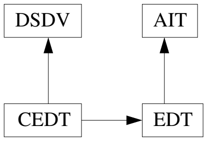


Fig. 2. Update dependency of the tables.

maintained by four tables: the EDSDV table, the Average Inter-meeting Time (AIT) table, the Estimated Delivery Time (EDT) table, and the Collective Estimated Delivery Time (CEDT) table. Each of these tables is of size $O(N)$ (a moderate transmission and memory requirement), where N is the network size.

The EDSDV table maintains the hop-count to the other nodes in the same cloud, while the hop-counts of the nodes not in the same cloud are ∞. Later, we will present the Economic DSDV (EDSDV), which requires each node to send incremental updates of sequence numbers only when topology is changed. It is an improvement of DSDV [13] where nodes frequently flood messages from which the nodes discover shortest paths. In DTNs where communication opportunities need to be discovered in a timely manner, EDSDV can substantially reduce the control overhead.

The AIT table records the average direct inter-meeting times (or waiting times) of the current node and all the other nodes in the networks. The AIT between two nodes is the averages of the periods of time between a disconnection and the consecutive establishments of a new connection. New inter-meeting times are weighted more. The AIT record between two nodes is ∞ if they met less than twice.

The EDT table maintains the minimal multi-hop transitivity inter-meeting time between the current node and the other nodes. For instance, if node A and node B have an AIT of 300 seconds, and B and C have an AIT of 200 seconds, then the EDT between node A and node C is at most 500 seconds. In the local mobility models two nodes should be close if they met recently. Also, node X is local to another node Y if X or some recent contact of X has a small average inter-meeting time with Y. Note that an AIT record being ∞ does not necessarily imply that that corresponding EDT is ∞.

The CEDT tables of the nodes in the same cloud are identical. Each CEDT record for a particular destination in the CEDT table equals the minimal record in the EDT tables of the nodes in the cloud. When a node moves into a cloud, its data messages whose destinations are in the cloud are first delivered by a multi-hop forwarding. Other messages that contribute to the minimal CEDT records are then forwarded to the nodes.

The update dependency of the above tables is shown in Figure 2. For instance, an arrow from table A to table B means that A's update is triggered by B's update. The updating of a node's EDT table is triggered by the update of its AIT table, whereas the update of a node's CEDT table is triggered by the updates of its neighbors' DSDV tables, EDT tables, or CEDT tables.

Algorithm 1 AROD

```
update tables AIT , EDSDV, EDT, and CEDT.
update ACKs and message vectors.
deliver destination-in-cloud messages.
while the node has message to send do
    calculate the priority of each messages.
    select and transfer a message with a probability proportional to its priority.
end while
send hello messages when all eligible messages are transmitted.
```

When nodes meet, they exchange the acknowledgements of the delivered messages and the message vectors of the messages that the nodes are storing before forwarding any data message.

The messages whose destinations are in the cloud are given priorities that are significantly larger than those of the other messages. The priorities of the destination-in-cloud messages are inversely proportional to the hop-count between its destination and the current node. The priority of a destination-out-of-cloud message is calculated based on the delivery probability (basically according to the CEDT record) and some fairness considerations. The priority of a message i destined for node d is defined as

$$Pr(i) = \frac{T_L - T_C}{E(d)} \cdot F(i) \cdot \frac{1}{T_L} \cdot \frac{1}{H},$$

where T_L is the original TTL of i, T_C is the creation time of i, $E(d)$ is an optimal expected delivery latency to d (which comes directly from the CEDT table), $F(i)$ is the ticket held by the replicas of i, and H is the hop count from the current node to the node contributing to the best CEDT record $E(d)$ for i's destination. $T_L - T_C$ is the remaining TTL of i, and $\frac{T_L - T_C}{L(d)}$ is the delivery probability of i based on the estimated delivery time. All messages have the same chance of being selected in the whole network since the $F(i)$s of all replicas of i add up to 1. $\frac{1}{T_L}$ gives all messages an equal chance of being selected during their lifetime. Finally, $\frac{1}{H}$ estimates the cost of forwarding the current message to the node contributing to $E(d)$.

The AROD algorithm is shown in Algorithm 1; without loss of generality, we describe the action a node takes when it comes in contact with other nodes.

B. Implementation Details

Each node sends "hello" messages to allow other nodes to detect it. Once a node detects "hello" messages from another node (neighbor), it maintains a contact record to store information about the neighbor, including the received table updates from the neighbor. Once no messages are received from a neighbor for a particular period of time, or a number of consecutive message-transfer failures occur with that neighbor, the contact with the neighbor is regarded as broken. To implement the above functions efficiently, we make simple modifications to the 802.11 MAC layer, such that the routing layer receives notifications directly from the MAC layer to indicate the connections/disconnections with neighbors. They also include the notifications about finishing sending a message, and whether a MAC layer ACK for a unicast is received. Having this notification from the MAC layer, we can implement a blocked transmission function with an ACK-received indicator that is returned in the routing layer, such that a reliable unicast is realized by rescheduling retransmissions when the failure of a previous transmission is indicated.

In DSDV [13], the DSDV tables (containing the ID, hop-count and the sequence number of each destination) are propagated frequently to discover shortest paths and broken paths. Although a timely distance vector update is important for the timely discovery of transmission opportunities in network scenarios where transmission opportunities are scarce, the propagation of the DSDV tables ($O(N)$) might eat up a considerable portion of the bandwidth. AROD uses an improved DSDV, Economic DSDV (EDSDV), in which a node sends a DSDV update only when its DSDV table is updated triggered by network topology changes. That is, incremental updates of an EDSDV table are sent only when necessary to minimize the control overhead. Hello messages, which are far less expensive, are used to discover neighbors.

Let's first briefly review DSDV. The basic idea is that each destination continuously floods newer sequence numbers in the networks, so that (1) each node knows the hop-count of a shortest path to the destination and the next hop on this shortest path, and (2) the last node in a cloud, which disconnects from the cloud where the destination is located, has the largest sequence number in its cloud. With this largest sequence number, its notification (of disconnection of the cloud and the destination), which is sent to the rest of the cloud, is trusted (and the count-to-infinity problem is solved).

The EDSDV table contains the following rows: dst-ID, next-ID, hop, dst-time. Unlike DSDV, a dst-time is functioned as the sequence number. When a node A chooses another node B as the the next hop node to a destination, A obtains the dst-time from B by decreasing it at a constant time T_{break}. T_{break} is the period of time without response between two nodes when the link between the nodes is considered broken. After receiving the deduced dst-time from B, A increases it by the time since the last hello message is heard from B. In this way, A's dst-time (the effective sequence number) will keep increasing, but will not be larger than that of B's. A node changes its next-ID (the id of the next hop node to the destination) when there is another route that either (1) contains a newer dst-time and its hop-count is not ∞ (which means the current route to the destination was broken.) or (2) the sequence number (dst-time) is equal to that of the current route, but it has a smaller hop-count.

EDSDV requires each node to store the most recent DSDV tables advertised by its neighbors, which requires a $D \times N$ memory space where D is the number of neighbors. It is obvious that EDSDV transmits fewer messages than DSDV in all circumstances. Also, there is no fluctuation [13] in EDSDV, which simplifies the algorithm and reduces the per DSDV table size. To implement incremental updates, an update time is maintained for each row in the EDSDV table. When sending updates to a neighbor, only the rows that are modified after the last update with the neighbor are sent. This method is also used for all of the other tables in AROD.

The AIT table contains five rows: dst-ID, last-con-time, last-discon-time, con-times, and AIT. Each time a node's connection with another node is broken, the corresponding last-discon-time is updated to the current time and con-times is increased by one. last-con-time and AIT are updated upon connection (except the first time). AIT (initially 0) is updated using the following equation:

$$AIT = (1 - factor) * AIT + factor * IT,$$

where $factor$ is the learning factor which equals $\max(\frac{1}{con-times}, 0.1)$ and IT is the last inter-connection time (the difference between the between the current time and the last-discon-time).

When the destination of a message is in the same cloud, the message is transmitted in the single-copy and multi-hop manner. In multi-hop transmissions, the logical ticket of a message is completely transferred to the copy sent to its receiver. That is, a message held by the sender has a ticket 0 and it can be removed from the sender's buffer. When a message is transmitted in a multi-copy manner, i.e., the message has an out-of-cloud-destination, the ratio of the ticket of the message retained in the sender and that of message sent to the receiver is proportional to the ratio of $\frac{1}{EDT}$ to $\frac{1}{CEDT}$.

C. Properties of AROD

AROD has properties identical to the routing schemes of different scenarios, as we will see in the following.

In a connected network, AROD uses multi-hop delivery, where a message will only be forwarded to the node that is closest to the destination in terms of hop-count. Since, in a connected network, the source and the destination are in the same cloud, the message will be forwarded in a multi-hop manner according to the EDSDV table. In each forwarding, the ticket of the message is completely transferred to the receiver, and thus the sender will not send the same message a second time. Consequently, a single copy of the message will be forwarded to the destination along the shortest path.

In a sparse mobile network with random mobility, AROD resembles *spray and wait* and performs binary spray. In random mobility, nodes have similar EDTs to any destination. A message will be forwarded in a multi-copy manner, where its ticket will be split among the copies from the sender and the receivers. After a certain number n of forwardings, the ticket of each copy will become approximately $(\frac{1}{2})^n$. As n becomes larger, the ticket, and consequently, the priority of the message, decreases exponentially. Then, the message will only have a very small chance of being forwarded when there are newer messages in the buffers and the transmission opportunity in the network is limited.

TABLE I
SIMULATION PARAMETERS

Parameters	**Values**
Field size	$1000 \times 1000(m^2)$
Number of nodes	50 or 10-50
Message rate	2 or 2-10 (msgs/s)
Buffer size	1MB
Data message size	1KB
Radio bandwidth	1Mb
Transmission range	100(m)
Message TTL	100 or 100-200(s)
Simulation time	500(s)
First message sending time	200(s)
Last message sending time	300(s)
Pause time in RWP	30-50(s)
Moving speed in RWP	20-60(m/s)
Radius of circles in CT	45/60/90(m)
Period of circles in CT	30/40/60(s)

In local mobility, AROD shows the trend to become more similar to a multi-hop delivery and less like multi-copy, mobility-assisted delivery. According to the definition of message priority, a message has a higher priority when the current node is in contact with a neighbor which has a smaller EDT for the message's destination. In this case, the message is more likely to be forwarded for having a larger priority. If the receiver's EDT to the destination is much smaller than that of the current node, a larger fraction of the ticket will be transferred to the receiver. The result of this is that, after forwarding, the sender's copy will have a very small priority and will not likely have another chance at being selected to forward in the future, whereas the receiver's copy, which has a ticket that is comparable to the original sender's ticket, will remain active.

In a DTN consisting of long term clouds, AROD broadcasts messages in the cloud if the capacity is plentiful. This maximizes the probability of the message being picked up by the mobile nodes from the other clouds. In the same cloud, most nodes likely have similar EDT to the out-of-cloud-destination of a particular message, and the fraction will be split at transmission, which allows the copies of the same message to be active before a number of copies are distributed.

The sum of the tickets of all the copies of a message is 1. This is generally true, except in situations where a sender, without receiving an acknowledgement, has no idea of whether a message has been forwarded to a receiver and thus can deduce from the local copy the ticket given to the sent copy. This situation arises when the receiver suddenly goes beyond the transmission range of the sender. This, however, is not a critical consistency problem in our routing protocol.

IV. SIMULATION

The following metrics are used in our simulation: (1) Convergence speed, (2) Delivery ratio, and (3) Delivery latency (delay). Two mobility models are used in our simulation: (1) the Random Waypoint (RWP) model, which is a representative of the random mobility, and (2) the Circular Trajectory (CT) model which is a representative of the local mobility. In the Circular Trajectory model, each node has a fixed circular trajectory that it travels at a constant speed. The snapshot of an example network with the Circular Trajectory model is shown in Figure 3, where trajectories are shown by dotted lines. Table I lists the basic parameters in the simulations.

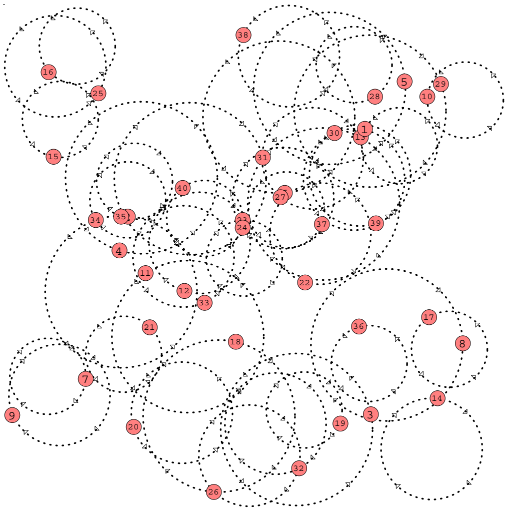


Fig. 3. An example network with the Circular Trajectory model.

A. Network Density

In the first experiment, we vary the number of nodes in the network from 10 to 50 nodes, in increments of 10 nodes in each step. Figure 4(a) shows that in both mobility models, the delivery rate increases as the network size increases. This result shows that AROD performs better in denser networks due to the adaptation of the multi-hop delivery, which saves bandwidth compared to the multi-copy delivery. Also, AROD performs better in RWP, which shows that increased mobility improves delivery rate.

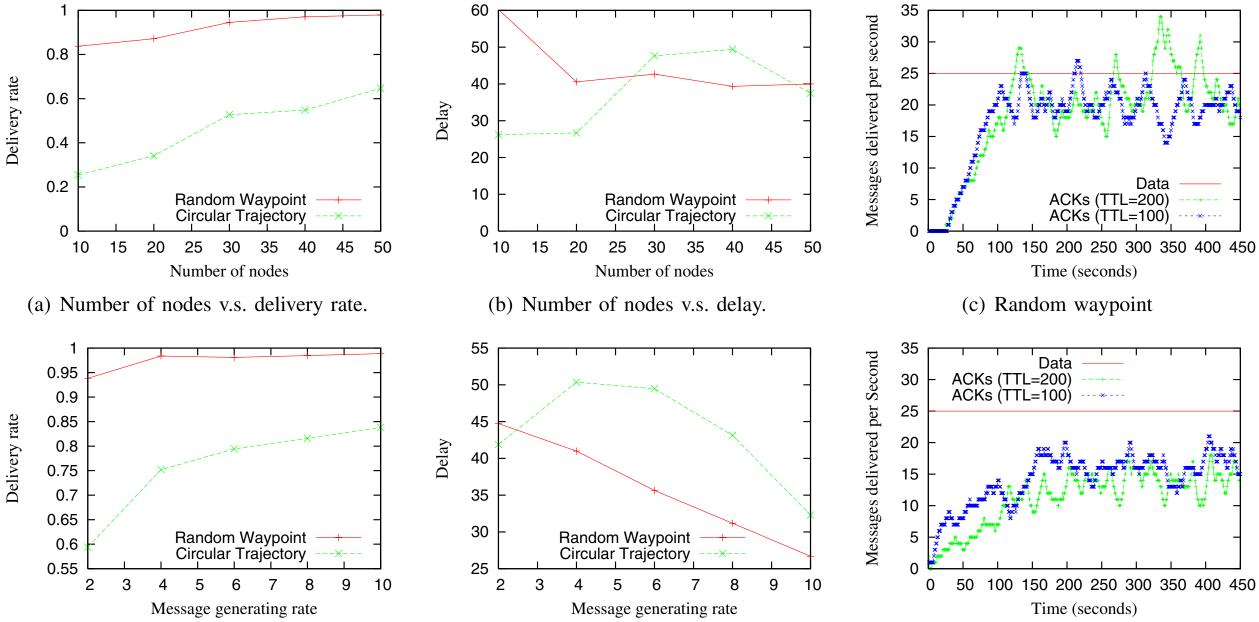
Figure 4(b) shows that the message delivery latency for RWP decreases as network density increases. The delay in the CT model follows the same trend. However, in the situation where the network density and the delivery rate are low, the CT model has a small delay since we only consider the delay of the delivered messages.

B. Message Generating Rate

In Figures 4(d) and 4(e), we vary the number of messages generated by each node per second from 2 to 10. In limited transmission opportunities, when fewer messages are generated, the number of transmissions shared by each message increases, and thus the delivery rate increases and the delay decreases. These figures show that AROD adaptively utilizes the bandwidth. Compared to spray-and-wait, it adaptively generates more (less) copies for the messages when message generating rate is lower (higher).

C. Convergence Speed and TTL

Figure 4(c) and Figure 4(f) show the convergence speed of AROD in the RWP model and the CT model respectively.



(a) Number of nodes v.s. delivery rate.

(b) Number of nodes v.s. delay.

(c) Random waypoint

(d) Message generating rate v.s. delivery rate.

(e) Message generating rate v.s. delay.

(f) Circular trajectory

Fig. 4. Delay v.s. number of nodes 4(a), 4(d) 4(b), 4(e) & message generating rate. Convergency speed 4(c), 4(f) (number of messages delivered v.s time & TTL).

AROD's EDT tables converge faster in the RWP model where each node meets a greater number of other nodes.

We can also observe in these figures that the delivery rate might increase or decrease as the TTL of the messages changes. In our simulation, the ACKs use the same TTL as the data messages. The longer the TTL, the longer a message stays in the buffer, giving it more chances to be delivered. On the other hand, with more ACKs, less bandwidth is available for the data messages since ACKs are delivered first.

V. Conclusion

In this paper we introduced a new routing scheme, Adaptive ROuting in Dynamic ad hoc networks (AROD) – a seamless integration of existing routing schemes. Simulation and discussion show that it has the desired performance with respect to delivery rate and delay. AROD is simple to implement and does not require configuration. In the future, we will continue to optimize AROD by the following improvements: (1) a reliable broadcast to reduce the bandwidth consumption, (2) an efficient ACKs dissemination, (3) using synthetic trace to evaluate the performance of AROD in more realistic scenarios, (5) using other routing protocols to compare to AROD.

VI. Acknowledgements

This work was supported in part by NSF grants ANI 0073736, EIA 0130806, CCR 0329741, CNS 0422762, CNS 0434533, CNS 0531410, and CNS 0626240.

References

[1] http://www.darpa.mil/ato/solicit/dtn/.

[2] A. Balasubramanian, B. N. Levine, and A. Venkataramani. DTN Routing as a Resource Allocation Problem, UM-CS-2006-033 October, 2006.

[3] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networking. In *Proc. of IEEE INFOCOM 2006*, 2006.

[4] S. Chen and K. Nahrstedt. A distributed quality-of-service routing in ad-hoc networks. *IEEE Journal on Selected Areas in Communications*, 17(8):547–560, August 1999.

[5] S. R. Das, C. E. Perkins, and E. E. Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. In *Proc. of IEEE INFOCOM*, pages 3–12, 2000.

[6] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. In *Proc. of ACM SIGCOMM*, August 2004.

[7] D. Johnson, D. Maltz, and J. Broch. DSR The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks. *Ad Hoc Networking, Addison-Wesley, Chapter 5*, pages 139–172, 2001.

[8] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In *Proc. of ASPLOS*, 2002.

[9] J. Leguay, T. Friedman, and V. Conan. Evaluating mobility pattern space routing for DTNs. In *Proc. of IEEE INFOCOM*, 2006.

[10] A. Lindgren, A. Doria, and O. Schelen. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):19–20, 2003.

[11] M. Mirco, H. Stephen, and M. Cecilia. Adaptive Routing for Intermittently Connected Mobile Ad Hoc Networks. In *Proceedings of the IEEE 6th International Symposium on a World of Wireless, Mobile, and Multimedia Networks (WoWMoM 2005).* IEEE press, June 2005.

[12] V. Naumov, R. Baumann, and T. R. Gross. An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces. In *Proc. of ACM MobiHoc*, 2006.

[13] C. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proc. of Conference on Communications Architectures, Protocols and Applications in conjection with SIGCOMM 94*, pages 234–244, 1994.

[14] T. Spyropoulos, K. Psounis, and C. Raghavendra. Efficient routing in intermittently connected mobile networks: The multi-copy case. In *ACM/IEEE journal of Transactions on Networking*, 2007.

[15] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-200006, Duke University, April 2000.