Performance Evaluation of Node Density-Based Adaptive Routing Scheme for Disruption Tolerant Networks[1]

Mooi-Choo Chuah*

Department of Computer Science & Engineering,
Lehigh University, PA 18015, USA
E-mail: chuah@cse.lehigh.edu
*Corresponding author

Peng Yang
Department of Computer Science & Engineering,
Lehigh University, PA 18015, USA
E-mail: pey204@lehigh.edu

Abstract: Traditional ad hoc routing protocols do not work in intermittently connected networks since end-to-end paths may not exist in such networks. Hence, routing mechanisms that can withstand disruptions need to be designed. A store-and-forward approach has been proposed for delivering messages in disruption tolerant networks. Recently, several approaches have been proposed for unicast routing in disruption-prone networks e.g. the 2-hop relay approach, delivery probability based routing, and message ferrying schemes. In our earlier paper, we have evaluated a combined multihop and message ferrying approach in disruption tolerant networks. In that paper, we assume that a special node is designated to be a message ferry. A more flexible approach is to let regular nodes volunteer to be message ferries when network dynamics mandate the presence of such ferries to ensure communications. Thus, in this paper, we design a node-density based adaptive routing (NDBAR) scheme that allows regular nodes to volunteer to be message ferries when there are very few nodes around them to ensure the feasibility of continued communications. Our simulation results indicate that our NDBAR scheme can achieve the highest delivery ratio (compared to other DTN routing approaches) in very sparse ad hoc networks that are prone to frequent disruptions.

Keywords: disruption tolerant networking; adaptive routing; node-density; 2 hop relay.

Biographical notes: M. Chuah received her PhD in Computer Science from University of California, San Diego. She is currently an Associate Professor in the Department of Computer Science and Engineering at Lehigh University. Her research interests are in heterogeneous network architecture design, resource management, network security, disruption tolerant network design, and adhoc/sensor networks. She runs the WiNS laboratory at Lehigh University.

P. Yang is a Ph.D. candidate in the Department of Computer Science and Engineering at Lehigh University. He received his MEng. in Computer Engineering from Nanyang Technological University, Singapore and a BEng. in Computer Engineering from the Huazhong University of Science and Technology, China. His current research focuses on architectural and protocol design for Delay/Disruption Tolerant Networks

[1] **Reference** to this paper should be made as follows: M. Chuah and P. Yang (2007) ‘Node Density-Based Adaptive Routing Scheme for Disruption Tolerant Networks’, *International Journal of Ad Hoc and Ubiquitous Computing*.

1 INTRODUCTION

Packet-switched network communication has been studied for decades. Important progress has been made in ensuring the robustness and scalability of the TCP/IP protocol suite. TCP/IP protocol suite is designed based primarily on the principles of end-to-end protocols and services. However, there are many scenarios in which an end-to-end connection is not guaranteed or even possible, and so an intermediary is needed, perhaps to translate between protocols or to provide temporary storage (e.g., in mail servers). In these cases, without such intermediaries, communication would fail. In other cases, communication may fail not because of a lack of instantaneous connection, but because the connection properties fall beyond the expected bounds (excessive round-trip-time or high packet loss probability). Solutions have been proposed to deal with some specific situations, e.g., using link layer retransmissions to deal with high packet loss probability in wireless environments (H. Balakrishnan et al., 1997). However, these solutions still do not work in situations where there are no end-to-end paths.

Some alternative solutions to deal with partitioned networks have emerged recently. For example, DakNet (A. Pentland et al., 2004) deploys physical transport devices, e.g., buses and motorcycles, to carry mobile access points between village kiosks and hubs with Internet connectivity so that the data carried by the physical transport devices can be automatically uploaded and/or downloaded when the physical transport devices are in the wireless communication range of a kiosk or a hub. Similar techniques are proposed in Magic Bike (Gittman, 2004) and Wi-Fi on Two Wheels (Arent et al, 2004). In the past year, considerable amount of research focusing on delay/disruption-tolerant networking and communications has been published (e.g. K. Fall et al., 2003; M. Chuah and L. Cheng et al., 2005; V. Cerf et al., 2004; W. Zhao et al., 2004). DieselNet (B. Burns et al., 2005) is a vehicular-based disruption tolerant network where connections between nodes are short-lived and occasional. A common approach used to address delays and disruptions is via the use of a store-and-forward mechanism similar to electronic mail (J. Klensin et al., 2001). This makes communication possible, even when an instantaneous end-to-end path does not exist.

Several routing schemes have been proposed for DTNs. They can be categorized into three categories: (i) using message ferries to connect partitioned nodes (W. Zhao et al., 2004), (ii) using history-based information to estimate delivery probability of peers and pass the message to the peer that can best deliver the message (A. Lindgren et al., 2003;J. Burgess et al., 2006), and (iii) using 2-hop relay forwarding schemes where a source can send multiple copies to different relay nodes and have the relay nodes deliver to the destination when they encounter the destination (Y. Wang et al., 2005; P. Juang et al., 2002).

In our earlier work (M.Chuah and P. Yang et al., 2005), we have evaluated the performance of a multihop routing scheme with custody transfer feature in a single domain DTN. We also have explored using message ferrying and high-power backhaul links for inter-domain message delivery. Our work revealed that in a single domain environment, even with the custody transfer feature, the delivery ratio drops when the nodes are sparsely connected. So, in this paper, we propose a node-density based adaptive routing (NDBAR) scheme that provides better performance than previous approaches. Our main contributions are (a) the design of an adaptive DTN routing scheme called NDBAR, (b) sensitivity analysis of tunable parameters of NDBAR, and (c) extensive comparisons of different DTN routing schemes in various scenarios e.g. different mobility models, different percentage of nodes that support DTN functionalities.

This paper is organized as follows: In Section 2, we summarized related work in the area of DTN routing schemes. In Section 3, we describe our node-density based adaptive routing scheme. In Section 4, we describe our simulation models. We also present and discuss our simulation results in different scenarios. For example, we study the impact of the node densities, mobility models, traffic models on the data delivery performance of the NDBAR-II scheme. We also compare the performance of the NDBAR-II scheme with the two-hop and multihop routing schemes. We conclude in Section 5.

2 RELATED WORK

Three categories of forwarding schemes have been proposed for DTNs. In the first category (W. Zhao et al., 2004; W. Zhao et al., 2005), the authors propose to use message ferries to gather data from stationary sources and deliver them to their destinations. For example in (W. Zhao et al., 2005), the authors assume that traffic demand between two nodes can be estimated. Then, they design routes for multiple ferries that can minimize the average data delivery latency. They also consider how nodes can be assigned to ferries based on different assumptions about ferry interactions. However, in military operations, sometimes traffic demands between nodes may not be easily estimated due to changes in battlefield situations. In addition, nodes that move can be message carriers themselves without having to resort to special message ferries.

In the second category (A. Lindgren et al., 2003; J. Burgess et al., 2006), the authors propose using history-based routing where each node maintains a utility value for every other node in the network, based on a timer indicating the time elapsed since the two nodes last encountered each other. These utility values which carry indirect information about relative node locations, get diffused through nodes' mobility. Nodes forward message copies only to those nodes with a higher utility for the message's destination. For example in (A. Lindgren et al., 2003), the authors propose a probabilistic metric called delivery predictability at every node for each known destination. This metric indicates how likely it is that a node will be able to deliver a message to each destination. The delivery predictability ages with time and also has a transitive property i.e. a node A that encounters node B which encounters node C allows node A to update its delivery predictability to node C based on its (A's) delivery predictability to node B and node B's delivery predictability to node C. In (A. Lindgren et al., 2003), a node will forward a message to another node it encounters if that node has a higher delivery predictability to the destination than itself. Such a scheme was shown to produce better performance than epidemic routing (A. Vahdat et al., 2000).

In the third category (Y. Wang et al., 2005; S. Jain et al., 2005), the authors propose using a 2-hop relay forwarding scheme where the source sends multiple copies (e.g. different erasure coding blocks) to different relaying nodes and the relaying nodes will deliver the copies they have to the destination node when they encounter the destination node. Again, such strategy will achieve small transmission overhead but may not enjoy high delivery ratio for messages with short deadlines.

3 NDBAR SCHEME

In (M.Chuah and P. Yang et al., 2005), we have evaluated the performance of a multihop routing protocol in a DTN scenario where 40 nodes were distributed over a geographical area of 1000x1000 to 4000x4000 m^2 (assuming a transmission range of 250m). We use a Dynamic Source Routing (DSR)-like multihop routing protocol (J. Broch et al., 1998) enhanced with the custody transfer feature (K. Fall et al., 2003). In DSR, a route request message is broadcasted by the source node to discover a route to a destination node. This route request message is relayed by intermediate nodes until it reaches the destination node. The destination node then sends a route reply back to the source node. In addition, the source node repairs the route when it receives a route error message from any intermediate node along the selected path. We enhance it to let intermediate nodes which have been selected as custodians to perform route repairs. Our simulation results indicate that when the node density drops below 4.4x10-6 (40 nodes over 3000x3000 m^2 which is equivalent to finding only one neighbor node within the transmission range of 250m), the delivery ratio drops significantly despite the custody transfer feature. Table 1 shows the simulation results for the scenarios with 40 nodes distributed over 3000x3000m^2 and 4000x4000m^2. We see that the achievable delivery ratio is only 54.3% and 18.3% respectively even with the custody transfer feature turned on. Thus, to improve on the delivery ratio, we design the node-density based adaptive routing (NDBAR) scheme where nodes can turn into message ferries when they detect that the node density around them drops below a certain threshold.

In the NDBAR scheme, we assume that each node periodically (e.g. every 20 seconds) broadcasts a neighbour discovery message to estimate n_d, the number of neighbours it has. When n_d drops below a certain threshold *K*, then that node will set a flag so that it will re-broadcast any future route-request message that it receives using high-power transmission. Any node that receives a high-power route request will take note of this fact, and will issue a high-power route reply when it hears from downstream nodes later. The high-power route reply message contains information about the location and speed of the node that replies. The previous-hop node that receives this reply will keep a record of this information so that if this route is chosen for packet delivery, the previous hop node will travel towards the next-hop node so that the data relay can be conducted using regular power transmission.

Table 1 *Performance of the multihop approach with custody transfer*

Simulation Area	Delivery Ratio	Avg Dly	95% Pkt Dly	Overhead	Hop count
3000x3000	54.6%	1829	3500	2.1	1.1
4000x4000	18.3%	932	1500	0.8	1

The NDBAR scheme consists of five components: (a) Local Node Density Estimation, (b) 2-hop Neighbor Contact Estimation, (c) Route Discovery, (d) Route Repair, and (e) Data Delivery.

(a) Local Node Density Estimation

Each node periodically broadcasts a *neighbor discovery* packet. On receiving a *neighbor discovery* packet, a node composes a *neighbor response* packet including this node's information (e.g. identifier, location, and velocity) and this node's 1-hop neighbor's information (e.g. the neighbor's identifier, contact probability, location, and velocity) and sends the *neighbor response* packet to the originator of the a *neighbor discovery* packet. Thus, each node can estimate the number of neighbors it has periodically, denoted as N_n . If a node's N_n drops below a threshold during a neighbor discovery period, the node sets a *sparsely connected flag*. Figure 1(a) shows the pseudo code for local node density estimation.

(b) 2-hop Neighbor Contact Estimation

Each node also maintains its contact probabilities with its 2-hop neighbors. The contact probability of a neighbor is set to 1 as long as a node, n_i , can receive neighbor response message from a neighbor, n_j , periodically. When n_i fails to hear a neighbor response message from n_j, then n_i decreases its contact probability with n_j by a factor of *a* periodically (since the neighbor discovery message is sent out periodically).

(c) Route Discovery

The source initiates a *route discovery* message if it doesn't have route information for a destination. The *route discovery* message includes the identifiers of source/destination and a forwarding list which will be used to record all intermediate nodes' tuples from the source to the destination. A node's tuple consists of two attributes: its node identifier and a *transmission flag* which indicates if the node uses high power transmission to rebroadcast the *route* *discovery* message. The source/intermediate node will check its *sparsely connected flag* before it broadcasts/rebroadcasts a *route discovery* message. If the flag is set, the source/intermediate node broadcasts/rebroadcasts the *route discovery* message using high power transmission; otherwise, it uses regular power transmission. Before the source/intermediate node broadcasts/rebroadcasts a *route discovery* message, it appends its node tuple into the forwarding list. On receiving a *route discover* message, a node checks its 2-hop neighbor table; if the node finds the destination in its 2-hop neighbor table, it composes a *route reply* message and forwards the *route reply* message along the reverse path to the source node; otherwise, it appends its node tuple into the forwarding list and rebroadcasts the *route discovery* message.

```
                         Local Density Estimation
Initialization:
 NN ← 0;                          //the number of 1-hop neighbors
 LOCAL_DENSITY ← 0;               //node density of neighborhood; 0 indicates low density,
                                  //1 indicates high density
 initialize(RESPONSE_TIMER);      //waiting timer for the response from neighbors
                                  //after broadcasting Neighbor_Discovery packet NDp
 initialize(BROADCAST_TIMER);     //timer for broadcasting Neighbor_Discovery packet NDp
 activate(BROADCAST_TIMER);       //activate the timer for the 1st broadcast

Upon expiration of BROADCAST_TIMER do
   localDensityDetection();

Upon reception of NDp do
   NRp ← composeNeighborResponse( );  //compose Neighbor_Response packet;
   send(NRp);                         // Send the Neighbor_Response message to the
                                      // sender of the Neighbor_Discovery message

Upon reception of NRp from a new neighbor node do
   NN ← NN + 1;
   update two-hop neighbor table;

Upon expiration of RESPONSE_TIMER do
   if NN ≤ DENSITY_THRESHOLD then
     LOCAL_DENSITY ← 0;
   else
     LOCAL_DENSITY ← 1;
   activate(BROADCAST_TIMER);

PROCEDURE localDensityDetection( )
   NN ← 0;
   NDp ← composeNeighborDiscovery();
   broadcast(NDp);
   activate(RESPONSE_TIMER);

PROCEDURE composeNeighborDiscovery()     //compose Neighbor_Discovery packet;
   create Neighbor_Discovery packet NDp;
   append own identifier in NDp;
   return NDp;

PROCEDURE composeNeighborResponse()      //compose Neighbor_Response packet;
   create Neighbor_Response packet NRp;
   append 1-hop neighbor information in NRp;
   append own identifier, location and velocity information in NRp packet;
   return NRp;
```

Figure 1(a) *Local Density Estimation*

The route reply message includes identifiers of source/destination, a reply list which is the reverse of the forwarding list and the current node's location. On receiving a route reply message, a node records the sender's location; in addition, the node checks the reply list to decide the next node to forward the route reply message and the transmission power it needs to use for such forwarding. Figure 1(b) shows the pseudo-code for the route discovery process.

(d) Route Repair

Custodian transfer feature (K. Fall et al., 2003) is turned on in our forwarding scheme. This means that after a node n_i receives a custodian acknowledgment from a downstream node, n_j, n_i can remove the acknowledged bundle from its storage. With this custody transfer feature enabled, one can use local route repairs when the route is broken as a result of node mobility. Let us assume that when bundles arrive at node n_4 in Figure 2, n_9 moves away. There are two ways whereby n_4 can repair the route: (a) n_4 can issue route request to find a route to the destination, or (b) n_4 can make use of the location and velocity information to travel closer to n_9 and not incur extra route discovery messages for local repair.

(e) Data Delivery

On receiving a *route reply* message, a source node generates a *delivery list* by reversing *reply list.* The *delivery list* is piggybacked in the *data* packet. On receiving a *data* packet, a node checks the *delivery list* for the next hop it is supposed to forward *data* packet to. Note that each intermediate node records the downstream node's location information provided in the route reply message. An intermediate node will act as a message ferry during data delivery if that intermediate node uses high power transmission to forward a *route discovery* to a downstream node. Note that a node that acts as a message ferry will do so only if either (a) it has received k data packets, or (b) the oldest data packet has been queued for more than 1000 seconds.

```
                         Route Discovery Process
Upon reception of route request for data packet delivery do
   RDp ← composeRouteDiscovery( );              //compose route discovery packet
   if LOCAL_DENSITY == 0 then
       broadcast RDp with high transmission power;
   else
       broadcast RDp with regular transmission power;

Upon reception of RDp do
   if RDp is duplicate
     drop RDp;
     return;
   else
    if lookUpReceiver( RDp)==TRUE then
      RRp ← composeRouteReply(RDp);
      forwardRouteReply(RRp);
      return ;

    if LOCAL_DENSITY == 0 then                  //append current node to forward list
      node.id ← LOCAL_ID ;
      node.power ← 1 ;                          //indicates high power transmission is used
      RDp.forward_list ← RDp.forward_list ∪ node;
      rebroadcast RDp with high transmission power;
    else
      node.id ← LOCAL_ID ;
      node.power ← 0 ;                          //indicates regular power transmission is used
      RDp.forward_list ← RDp.forward_list ∪ node;
      rebroadcast RDp with regular transmission power;

Upon reception of RRp do
   if RRp.destination == LOCAL_ID then
    route ← reverse(RRp.reply_list);
    use route to delivery data packets in buffer;
   else
    forwardRouteReply(RRp);

PROCEDURE lookUpReceiver(RDp)                    //look up receiver in local area
   look for RDp.destination in the 2-hop neighbor table;
   if found destination then
     append downstream nodes to the destination to RDp.forward_list;   //note that node.power ← 0
     return TRUE;
   else
     return FALSE;

PROCEDURE composeRouteReply(RDp)
   create Route_Reply packet RRp;
   RRp.destination ← RDp.source;
   RRp.source ← LOCAL_ID;
   RRp.reply_list ← reverse(RDp.forward_list);
   return RRp;

PROCEDURE composeRouteDiscovery()
   Create Route_Discovery packet RDp;
   RDp.source ← LOCAL_ID;
```

Figure 1(b) *Route Discovery*

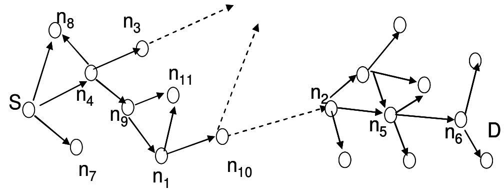

Figure 2 *NDBAR scheme*

In Figure 2, we illustrate how the NDBAR scheme works via an example. The source node, S, broadcasts a route request at regular power. Node n_7 , n_4 , and n_8 hear this route request and will re-broadcast the route request using regular power. Node n4's rebroadcast is heard by nodes n_9 , n_3 , and n_8 . Similarly, node n_9 's rebroadcast is heard by nodes n_{11} and n_1 . Node n_{10} realizes that its observed number of neighbors is 1 (assume K is set to 1.5). Thus, upon hearing the route request from node n1, n10 issues a high-power route request which reaches node n_2 . Node n_2 takes note that it receives a high-power route request from node n_{10} and rebroadcasts using regular power since the number of neighbors it observes exceeded K. This goes on until the route request reaches node D which is the destination. When n_2 receives a regular-power route reply from n_5 , it issues a high-power route reply to node n_{10} after attaching information regarding its (n_2) location and velocity. Node n_{10} relays this route reply back to S via node n_1 using regular-power route reply after recording n_2 's location and velocity information. Since we assume that the data transmission rate is higher than the route request rate, we design the NDBAR scheme such that the node with low connectivity delivers the data packets via message ferrying. Thus, if this route is chosen, then when node n_{10} receives data packets from node n_1, it will travel towards node n_2 until it (n_{10}) is close enough to deliver the data packets via regular power transmission to node n_2 . Note that n_{10} can decide when it wants to move (e.g. after receiving several packets) towards n_2 depending on the speed of n_2 and the message rate that n_{10} receives from n_1 .

4 PERFORMANCE EVALUATION

We implemented the NDBAR scheme, the enhanced multihop and the two hop routing schemes using NS-2 network simulation package (http://www.isi.edu/nsnam/ns/) and the simulation results are presented in this section. We also implement the custody transfer feature described in (K. Fall et al., 2003; M. Chuah and P. Yang et al., 2005). The custody transfer feature works as follows: accepting a message with custody transfer amounts to promising not to delete it until it can be reliably delivered to another node providing custody transfer or it arrives at the destination. Nodes holding a message with custody are called custodians. Normally, a message has a single custodian (referred to as sole custody) but in some circumstances, more than one custodian owns a message or message fragment (referred to as joint custody). We do not simulate message fragmentation so there is no joint custodian in our simulation. The number of packets queued before a node starts acting as a message ferry, k, is set to 10.

IEEE 802.11 MAC is used. The transmission range of the radio is set at 250m and the high power transmission is assumed to extend the transmission range to 500m. Each node is assumed to have 200 buffers. There are 10 CBR flows and unless otherwise indicated, each flow has a packet rate of 1 every 4 seconds. The packet size is 512 bytes. The source and the destination of each flow is randomly selected. The performance metrics we use are:

- Packet delivery ratio (PDR) which is the number of packets that are correctly delivered to the destination over the number of unique data packets sent by the source.
- End-to-end delivery latency which is the time it takes to delivery a data packet. We consider both the average and the 95 percentile values.
- Hop count which is the average number of hops it takes for a data packet to arrive at the destination.
- Transmission overhead [Y. Wang et al., 2005] which is defined as the number of transmitted bytes over the number of generated bytes. The transmission bytes include the routing overhead messages, custody transfer request and acknowledgment messages. Custody transfer and acknowledgment messages are assumed to be 35 bytes each.

We conducted several sets of experiments. Unless otherwise stated, the aging factor is set to 0.8 and the neighbor discovery interval is set to 20 seconds. For mobility, we either use the random waypoint model (RWP) (J. Broch et al., 1998) or the ZebraNet mobility model (Y. Wang et al., 2005). For the random waypoint model, each node moves towards a randomly picked destination at a constant speed. Once the destination is reached, another destination will be randomly chosen and the node will start moving towards the new destination after a certain pause time. This behavior is repeated for the whole duration of the simulation. In our simulation, unless otherwise stated, the regular node's speed is chosen uniformly between zero and 5 m/s. If the node needs to act as a message ferry, then it uses a speed of 15 m/s unless otherwise stated. For ZebraNet (Y. Wang et al., 2005) movement, we scale the node positions to fit into the geographical area used in our scenarios. We also scale the sampling time to be 8 seconds rather than 8 minutes. All the reported delay values in this paper are in seconds.

4.1 Impact of Node Density

In our first set of experiments, we have 40 nodes distributed randomly over (a) 3000x3000 m^2, (b) 4000x4000 m^2, and (c) 5000x5000 m^2. Table 2 tabulates the results. It shows that NDBAR can significantly improve the delivery ratio but it comes at the expense of transmission overhead. To improve the delivery ratio from 54.6% (refers to Table 1) to 96.2% (refers to Table 2) for the 3000x3000m^2 scenario, one has to pay a transmission overhead of 17.5. It is almost impossible to deliver packets for case (c) using only multihop routing with custody transfer feature turned on but the NDBAR scheme can achieve a delivery ratio of 81.5% using a transmission overhead of 3.1.

Table 2 Performance of NDBAR scheme (RWP)

Simulation Area	Delivery Ratio	Delay	Hop count	Overhead
3000x3000	96.2%	818 sec	7.2	17.5
4000x4000	95.5%	1688 sec	5.3	12.4
5000x5000	81.5%	3455sec	2.8	3.10

To reduce the transmission overhead, we consider a variant of NDBAR (referred to as NDBAR-II). Each node is assumed to exchange information of its 1-hop and 2-hop neighbors with its immediate neighbors. A source node sends the data packet directly to a node that can reach the destination without going through the route request procedure. Otherwise, it starts a neighbor relay expiry timer, w (set to 2000s). If a packet can be delivered to its destination via neighbor relaying before *w* expires, then a route request will not be issued. During the neighbor relaying period, a node will send the data packet to a neighbor with the highest contact probability to the destination. Otherwise, a route request will be issued by the node which receives the data packet. If the message or route request is received by a node that does not have enough neighbors, then the node is allowed to issue a high power route request message. Each node maintains its contact probabilities with its 1 hop and 2 hop neighbors. Let us denote node *i*'s contact probability with node *j* as P_j^i. P_j^i is updated as follows: node *i* periodically broadcasts a neighbor discovery message; if node *i* hears a response back from node *j*, then P_j^i is set to 1; otherwise the existing P_j^i value decays by a factor □ (set to 0.8 in our experiments) periodically. We refer to this variant as the NDBAR-II scheme. We illustrate NDBAR-II in Figure 3. The source node S attempts to deliver the packet to destination D initially via 2-hop relaying until the packet reaches node n_3 where the number of neighbors drops below the threshold so a high power route request is issued.

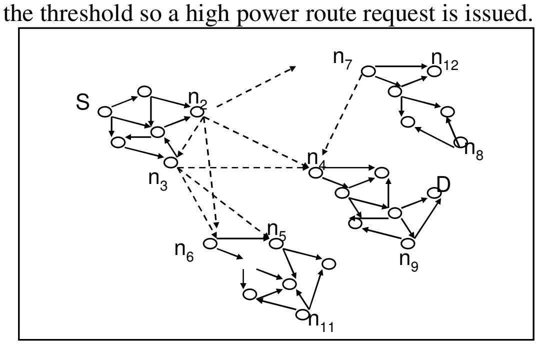

Figure 3 *Enhanced NDBAR scheme (NDBAR-II)*

The simulation results for the NDBAR-II scheme is shown in Table 3. Our results show that the NDBAR-II scheme can achieve relatively high delivery ratio (over 93.9% in 3000x3000m^2 scenario) but with much reduced transmission overhead (decreases from 17.5 to 7.4). This comes at the cost of increasing the 95% message delivery latency. The transmission overhead will reduce further if higher packet generation rate is used.

4.2 Sensitivity Analysis of Parameters of NDBAR-II

In this section, we investigate how different tunable parameters e.g. different aging factors, ferrying speeds, neighbor discovery intervals affect the delivery performance of the NDBAR scheme (i.e. NDBAR-II). We use a network scenario of 40 nodes distributed over an area of 4000x4000m^2. The nodes are moving according to random waypoint model and with a maximum speed of 5 m/s. Again 10 CBR flows with each flow generating 0.25 pkt/sec.

Table 3 *Performance with NDBAR-II scheme (RWP)*

Simulation Area	Delivery Ratio	Avg Delay	95% packets delay	Hop count	Overhead
1000x1000	99.0%	203	700	2.4	5.3
2000x2000	96.4%	1470	4000	2.3	4.3
3000x3000	93.9%	2465	4327	5.4	7.4
4000x4000	92.4%	2654	5123	4.3	7.2
5000x5000	79.6%	3921	8873	3.7	7.1

Table 4 shows the results with different aging factors while Table 5 shows the results with different ferrying speeds. From Table 4, we see that having smaller aging factor allows the system to forget outdated information faster and hence improves the delivery performance slightly. Table 5 shows that having a faster ferry speed improves on the data delivery performance. When the ferrying speed increases from 15 m/s to 30 m/s, the delivery ratio improves from 92.4% to 93.6%. The average delay drops from 3176 seconds to 3121 seconds.

Table 4 *Impact of different aging factors on data delivery performance*

Aging factor	Delivery Ratio	Avg Delay	95% Delay	Transmission Overhead
0.8	92.4%	3176	5123	7.2
0.6	93.1%	3065	4808	7.2
0.4	93.7%	2955	4226	6.5
0.2	93.9%	2946	4201	6.5

Table 5 *Impact of different ferrying speeds on data delivery performance*

Ferry Speed	Delivery Ratio	Avg Delay	95% Delay	Transmission Overhead
15	92.4%	3176	5123	7.2
20	92.6%	3156	5003	7.2
25	93.2%	3132	4883	7.2
30	93.6%	3121	4823	7.2

We also investigate the impact of changing the neighbor discovery intervals. Table 6 shows the data delivery results we obtained. The results indicate that setting the neighbor discovery interval at 20 seconds is appropriate for the scenario we study since it achieves relatively high delivery ratio with reasonable transmission overhead.

For the remaining simulation experiments in this paper, we set the neighbor discovery interval to 20 seconds and the aging factor to 0.8.

Table 6 *Delivery Performance with different neighbor discovery intervals*

Neighbor Discovery Interval (seconds)	Delivery Ratio	Avg Delay	95% packets delay	Overhead
5	92.9%	2112	3112	9.1
10	92.9%	2123	3123	7.6
20	92.4%	2654	5123	7.2
30	88.5%	2456	4976	6.2

4.3 Comparison with 2-hop erasure-coding and multihop schemes

In this section, we compare the routing performance of 2-hop erasure-coding relaying scheme (Y. Wang et al., 2005; M. Chuah and P. Yang et al., 2005), the multihop routing scheme with custody transfer (K. Fall et al., 2003), and the NDBAR-II scheme. We simulated the scenarios tabulated in Table 7.

Table 7 *Simulation Parameters*

Parameter	Value
Simulation area	1000x1000, 2000x2000, 3000x3000, 4000x4000
Simulation time	10000 seconds
Traffic pattern	10 CBR flows with a packet size of 512 byte
Mobility model	RWP with maximum speed equal to 5m/s, Zebranet Mobility Pattern

Figures 4,5,6&7 plot the delivery ratio, the average delay, 95%tile delay and overall overhead results for the 2-hop erasure-coding relay approach, the multihop approach and the NDBAR approach respectively with movements based on the random waypoint mobility model. Figures 8,9,10 & 11 plot similar results for the three schemes using Zebranet mobility model (Y. Wang et al., 2005).

Figures 4 shows that the 2-hop approach performs poorly when the random waypoint mobility model is used. It achieves only 78.4% in the $1500x1500m^2$ scenario and only 41.9% in the 3000x3000 m^2 scenario. The delivery ratio with multihop approach is very good until the node density drops below $4.4x10^{-6}$. NDBAR achieves very high delivery ratio at all scenarios. The sudden drop in average delay for the multihop approach in Figure 5 can be explained as follows: at very low node density, the delivery ratio for the multihop approach drops so much that only very few packets that can be easily forwarded were successfully delivered. Except for NDBAR, the transmission overhead for the other three approaches in general drops gradually with decreasing node density initially and drops suddenly when the node density becomes really sparse. Again, this effect is caused by the inability of the three routing approaches to deliver messages in very sparse environment. NDBAR however adapts itself to deliver messages in a very sparse environment by exploring different delivery options and hence incur higher transmission overhead in most cases.

Our results in Figure 8 reveal that the 2-hop approach provides relatively good performance if ZebraNet mobility model is used until the node density drops below $4.4x10^{-6}$ ($3000x3000m^2$ scenario) when the delivery ratio becomes 76.2%. The results in both sets of plots clearly show that the NDBAR-II scheme achieves the best delivery ratio in very spare networks with reasonable transmission overhead. It also shows that the NDBAR-II scheme is flexible enough to handle different mobility models.

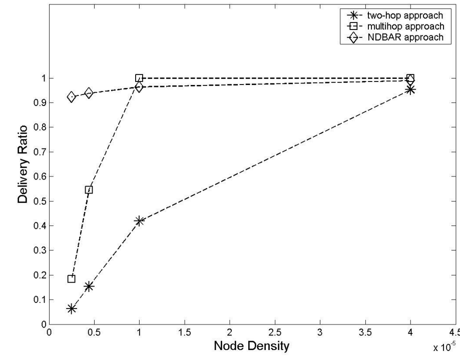

Figure 4 *Delivery Ratio using RWP mobility model*

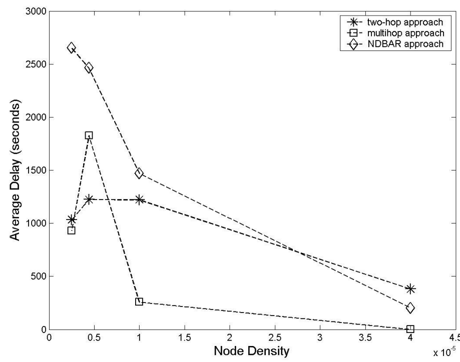

Figure 5 *Average Delay using RWP mobility model*

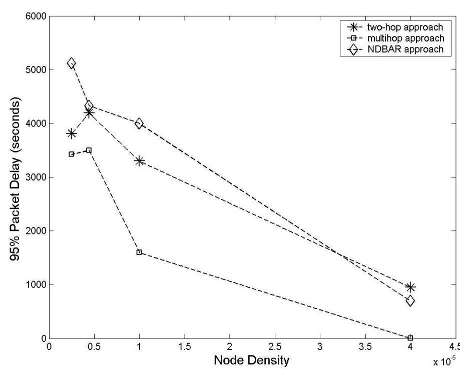

Figure 6 *95%tile delay using RWP mobility model*

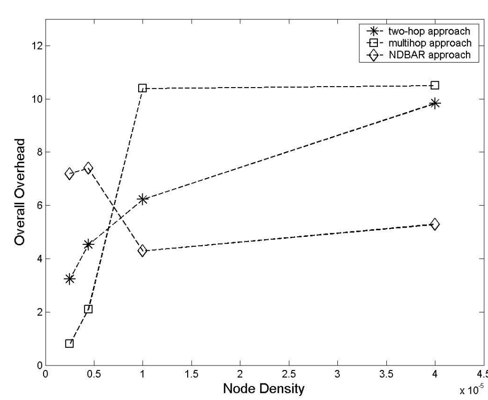

Figure 7 *Overall Overhead using RWP mobility model*

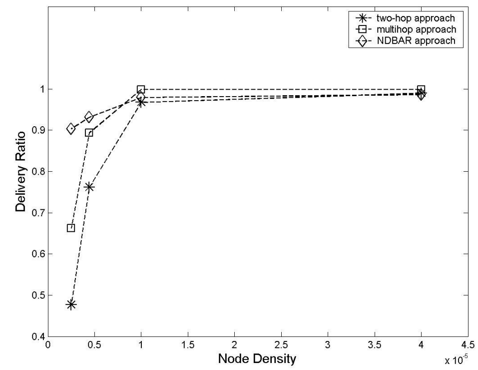

Figure 8 *Delivery Ratio using Zebranet mobility model*

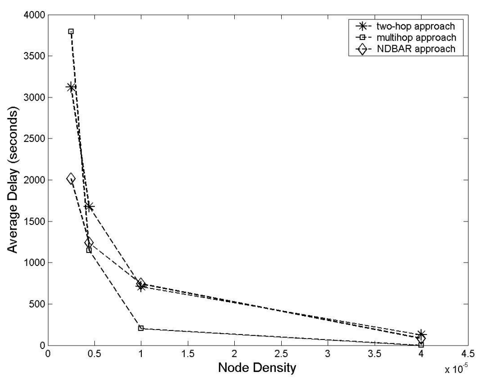

Figure 9 *Average Delay using Zebranet mobility mode*l

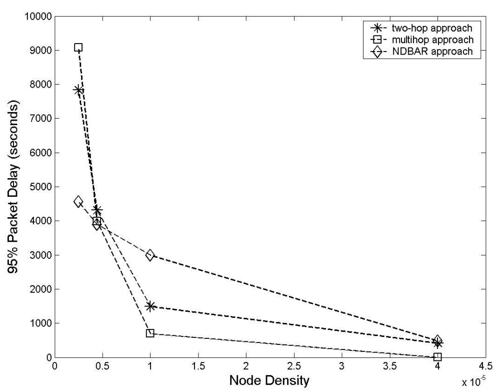

Figure 10 *95%tile Delay using Zebranet mobility model*

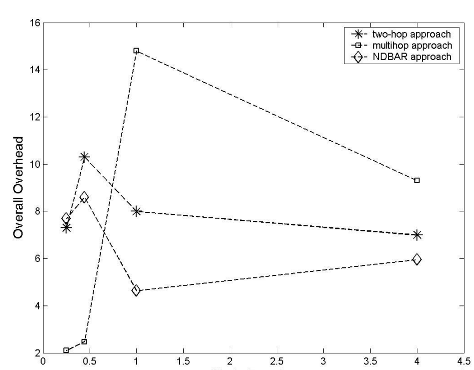

Figure 11 *Overall Overhead using Zebranet mobility model*

4.4 Impact of Traffic Model

We next evaluate the impact of traffic model on the data delivery performance with the NDBAR-II scheme and random waypoint mobility model.

In earlier sections, we use one-way CBR flows with each flow generating 0.25 pkt/sec as our traffic model. In this section, we use a network scenario with 40 nodes distributed over 4000x4000m^2 but uses a bidirectional traffic model described in (W. W. Brown and T. Krout. 2005; J. Hsu et al., 2003). The source sends a message to the destination. Upon receiving the message, the destination will respond with another message. The random waypoint mobility model is used in this set of experiments. We evaluate the end-to-end delay of the bidirectional message flows.

Table 8 tabulates our simulation results. Compared to the one-way flow results we obtained earlier, the 95% bidirectional message delay is only about 10% higher than the 95% unidirectional message delay when the node density is above 4.4x10^{-6} but it almost doubles when the node density decreases to 4.4x10^{-6} and it triples when the node density drops to 2.5x10^{-6}. The delivery ratio has dropped to 85.3% with bidirectional flows when the node density is 2.5x10^{-6} but it is still relatively high. The transmission overhead improves since the messages in the reverse direction do not have to incur extra route discovery overhead.

Table 8 *Delivery Performance with bidirectional flows using NDBAR-II (RWP)*

Simulation Area	Bidirectional-Delivery Ratio	Avg Delay	95% Pkt Delay	Overhead	Hop Count
1000x1000	99.0%	260	734	3.3	4.8
2000x2000	96.4%	533	4417	2.7	4.3
3000x3000	91.0%	4576	8643	9.1	9.6
4000x4000	85.3%	6414	15435	6.6	7.4

4.5 Impact of different percentage of DTN nodes

In some scenarios, not all nodes may support DTN functionalities. Nodes that do not support DTN functionalities can not be message ferries. Thus, in this section, we investigate the impact of having different percentage of the nodes supporting DTN functionalities on the data delivery performance. Table 9 tabulates our results when different percentages of nodes support DTN functionalities.

From Table 9, we see that the delivery ratio degrades from 92.4% when all nodes support DTN functionalities to 71.5% when only 25% of the nodes support DTN functionalities.

Table 9 *Impact of having different percentage of nodes supporting DTN functionalities*

Percentage of DTN nodes	Delivery Ratio	Avg Delay	95% Delay	Transmission Overhead
100%	92.4%	3176	5123	7.2
75%	89.7%	4567	7853	6.4
50%	87.3%	5396	9035	5.7
25%	71.5%	6245	9055	6.8

4.6 Impact of Node Movement Speeds

In this section, we explore how the different node movement speeds affect the data delivery performance. We use a network scenario with 40 nodes distributed over 4000x4000 m with 10 CBR flows each generating 0.25 pkt/sec. The sources and the destinations of these 10 flows are randomly chosen. The results are tabulated in Table 10. From Table 10, we can see that the delivery ratio does not change much with increasing node speeds but the average delay and the 95 percentile delay increases with increasing speed. The higher delay can be attributed to the fact that more route discovery time is incurred with more frequent partitions. It is interesting to note that the overall transmission overhead decreases slightly with increasing node speeds. Since the source can encounter more nodes with faster movement and hence deliver more packets via the 2-hop routing approach more frequently, fewer control messages are spent to discover routes.

Table 10 Delivery performance with different node speeds

Speed Rang & Average Speed	Delivery Ratio	Avg Delay	95% packets delay	Overhead
(0,5) 2.4	92.4%	2654	5123	7.2
(5,10) 7.0	92.0%	2854	6323	6.5
(10,15) 11.6	92.2%	3012	7156	5.8

4.7 Impact of Traffic Load

In this section, we investigate how the NDBAR scheme performs when the traffic load changes. We use a network scenario with 40 nodes distributed over $4000x4000m^2$ and the nodes move according to the random waypoint model. There are 10 CBR flows with random sources and destinations. The data generation rate of each flow is varied from 0.25 pkt/sec to 2 pkt/sec. Figures 12, 13, 14 & 15 plot the results of the delivery ratio, the mean & 95 percentile delay, and the overall overhead respectively. Our results indicate that the NDBAR scheme maintains high delivery ratio as the traffic load increases. The price to pay is higher overall overhead. However, the overall overhead incurred by the NDBAR scheme decreases with increasing load since more packets can be forwarded using the same route.

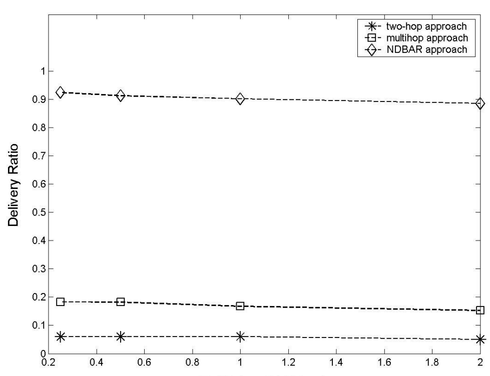

Figure 12 *Delivery Ratio vs. Traffic Load*

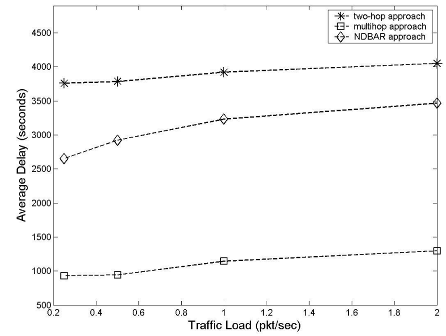

Figure 13 *Average Delay vs. Traffic Load*

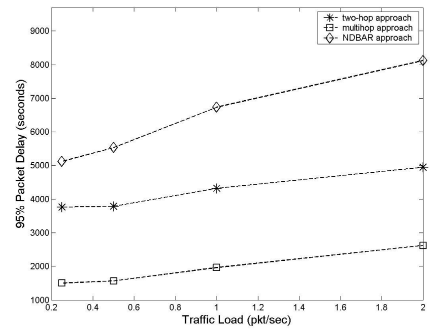

Figure 14 *95 Percentile Delay vs. Traffic Load*

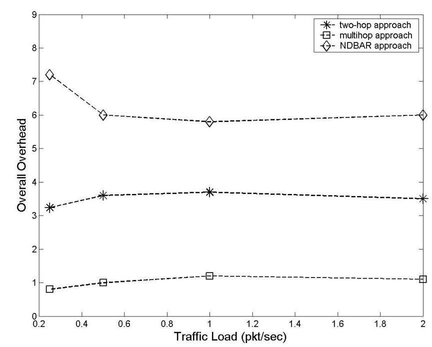

Figure 15 *Overall Overhead vs. Traffic Load*

5 CONCLUSION

Our simulation results indicate that the multihop with custody transfer approach provides better delivery ratio than the 2-hop erasure-coding approach but incurs higher transmission overhead. However, the results also indicate that both the 2-hop erasure-coding relay approach and the multihop with custody transfer approach fail to provide reasonable delivery ratio when the node density is lower than $4.4x10^{-6}$ (with a regular transmission range of 250 m). Thus, in this paper, we have designed a new routing scheme called the Node Density Based Adaptive Routing (NDBAR) scheme for DTN environment. Our scheme allows a node to decide when it will function as a message ferry to deliver data packets using its locally observed neighbor density information. The enhanced NDBAR scheme (NDBAR-II) scheme achieves the best delivery ratio. Our simulation results indicate that it can achieve more than 90% delivery ratio using ZebraNet mobility model and 92% delivery ratio with a node density of $2.5x10^{-6}$ when the transmission range is 250m. With bidirectional flows, our NDBAR-II scheme can still achieve 85.3% under the same network conditions. The 95 percentile delay for bidirectional traffic is only 10% higher than the one-way traffic. Our evaluations also indicate that the NDBAR-II scheme can achieve a delivery ratio of 71.5% (87.3%) with only 25% (50%) of the nodes supporting DTN functionalities. Recall that the 2-hop routing scheme can only achieve a delivery ratio of 18% in this scenario. The above results assume the average node speed is 2.4 m/s. We also found that the delivery ratio of the NDBAR-II scheme remains high (92.2%) with nodes moving at an average speed of 11.6 m/s and at a high total traffic load of 20 pkts/sec (10 flows with 2 pkt/sec each).

In this paper, we assume that the node movements can be controlled. We also have simulation results for a variant of NDBAR where the intermediate node that does not has enough neighbors performs random walk (rather than acting as message ferry). Our results show that their performance is slightly lower than NDBAR but still reasonably close. We intend to implement the NDBAR-II scheme and evaluate its performance in a reasonable size test-bed.

ACKNOWLEDGEMENT

This work is sponsored by Defense Advanced Research Projects Agency (DARPA) under contract W15P7T-05-C-P413. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA. This document is approved for public release, unlimited distribution.

REFERENCES

A. Pentland, R. Fletcher, and A. Hasson, “DakNet: Rethinking connectivity in developing nations”, *IEEE Computer*, 37(1):78-83, January 2004.

H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, “A comparison of mechanisms for improving TCP performance over wireless links”, *IEEE/ACM Transactions on Networking*, 5(6):756-769, December 1997.

K. Fall, “A delay-tolerant network architecture for challenged Internets”, *Proceedings of SIGCOMM’03*, August 2003.

M. Chuah, L. Cheng, and B. D. Davison, “Enhanced Disruption and Fault Tolerant Network Architecture for Bundle Delivery (EDIFY)”, Proceedings of IEEE Globecom, 2005

J. Broch etc, “A performance comparison of multihop wireless adhoc network routing protocol”, Proceedings of Mobicom, 1998.

J. Klensin, Editor, “Simple mail transfer protocol”, *IETF RFC* 2821, April 2001.

R. Viswanathan, J. Li, and M. C. Chuah, “Message Ferrying for Constrainted Scenarios”, poster, *Proceedings of WoWMoM*, 2005.

W. Zhao and M. H. Ammar, “Message Ferrying: Proactive Routing in Highly Partitioned Wireless Adhoc Networks”, *Proceedings of the IEEE Workshop on Future Trends in Distributed Computing Systems*, Puerto Rico, May, 2003.

V. Cerf et al, “Delay-Tolerant Network Architecture”, Internet Draft, draft-irtf-dtnrg-arch-02.txt, July 2004.

B. Burns, O. Brock and B. N. Levine, “MV Routing and Capacity Building in Disruption Tolerant Networks”, *IEEE Infocom*, March 2005.

W. Zhao, M. Ammar, and E. Zegura, “A message ferrying approach for data delivery in sparse mobile ad hoc networks”, *ACM Mobihoc*, May 2004.

W. W. Brown and T. Krout, “Future Performance Expectations for Mobile Wireless Communication Networks”, preprint, 2005

J. Hsu, S. Bhatia, M. takai, R. Bagrodia, M. J. Acriche, ”Performance of Mobile Adhoc Networking Routing Protocols in Realistic Scenarios”, Milcom 2003.

K. Fall, W. Hong and S. Madden, “Custody transfer for reliable delivery in delay tolerant networks”, IRB-TR-03-030, July 2003.

Y. Wang etc, “Erasure-Coding Based Routing for Opportunistic Networks”, Proceedings of Sigcom WDTN workshop, 2005.

P. Juang, etc, “Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet”, Proc ASPL0S, Oct 2002.

A. Lindgren, A. Doria, O. Schelen, ”Probabilistic routing in intermittently connected networks”, Sigmobile, Mobile Computing and Communications Review, Vol 7(3), pp 19-20, 2003.

S. Jain, M. Demmer, R. Patra, K. Fall, “Using Redundancy to cope with Failures in a Delay Tolerant Network”, Proceedings of Sigcomm 2005.

A. Vahdat, D. Becker, “Epidemic Routing for partially connected adhoc networks”, Tehnical Report CS-200006, Duke University, April 2000.

M.Chuah, P. Yang, B. Davison, L. Cheng, “Performance Comparison of Unicast Delivery Schemes for DTNs”, CSE Technical Report, Lehigh Univeristy, Oct, 2005.

J. Burgess, B. Gallagher, D. Jensen, B.N. Levine, “MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networking”, to appear in Proceedings of IEEE Infocom, April, 2006.

W. Zhao, M. Ammar, and E. Zegura, “Controlling the mobility of multiple data transport ferries in a Delay-Tolerant Network”, Proceedings of IEEE Infocom, April, 2005.

WEBSITES

Y. Gitman, “Magic Bike: Wireless Internet”, http://www.magicbike.net/

L. Arent, G4techTV, Tech-Live, “Wi-Fi on Two Wheels”, http://www.g4techtv.com/techtvvault/features/47657/Wi Fi_on_Two_Wheels.html, May 2004

UCB/LBNL/VINT, “The Network Simulator ns-2”, Online at http://www.isi.edu/nsnam/ns/