# Darpa Contract # W15P7T-05-C-P413
# Deliverable #3(a)
# Store-and-Forward Performance in DTNs:
## Impacts of Custody Transfer, Message Ferry, Mobility Models

Lehigh University

Oct 31, 2005

*Abstract*—**Delay and disruption tolerant networks have been proposed to address data communication challenges in network scenarios where an instantaneous end-to-end path between a source and destination may not exist, and the links between nodes may be opportunistic, predictably connectable, or periodically-(dis)connected. In this report, we first give a brief summary of our EDIFY architecture and then describe the system operations e.g. neighbor discovery, custody transfer in EDIFY. Next, we describe the routing schemes that have been proposed for intragroup communications and discuss how intergroup route discovery takes place. Later, we describe the various performance studies we performed for different DTN scenarios e.g. DTNs with different node densities, DTNs with different mobility models. In addition, we also study a DTN scenario where message ferry is used. Our results indicate that (a) the store-and-forward and custody transfer concepts have significantly improved the delivery ratio in a sparsely connected network, (b) in very sparse network, message ferries are required to enable communications. We conclude with some discussions of the future work that we intend to perform in the remaining time of the project.**

***Keywords-disruption tolerant networks; custody transfer; route discovery, message ferry***

## I. INTRODUCTION

Packet-switched network communication has been studied for decades. Important progress has been made in robustness and scalability in the TCP/IP protocol suite based primarily on principles of end-to-end protocols and services [end-to-end]. 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 [4]. However, these solutions still do not work in situations where there are no end-to-end paths. DakNet [3] 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 [1],[2]. In the past year, considerable amount of research focusing on delay/disruption-tolerant networking and communications has been published (e.g., [13],[15]). Diesel [14] is a 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 [11]. This makes communication possible, even when an instantaneous end-to-end path does not exist

In [5], Fall describes an architecture for delay tolerant networking that implements much of what we have described. It proposed the idea of topological regions connected by gateways, which were responsible for storing messages in non-volatile storage to provide for reliable delivery. End-point addressing in his scenario consisted of a region name used for inter-region routing and a locally-resolvable name for intra-region delivery. More recently, we have proposed an enhanced disruption-tolerant network architecture called EDIFY (Enhanced Disruption and Fault Tolerant Bundle Delivery) [6]. Our approach builds on many ideas from Fall, but adds support for multiple, overlapping name spaces and node and group mobility.

In this report, we first give an overview of our enhanced DTN architecture. Then, we describe the store-and-forward and custody transfer features that will be used in a DTN. Next, we describe the different forwarding schemes that have been proposed for DTNs and discuss the pros and cons of each approach. Then, we describe different experiments we have done to understand the impacts of the store-and-forward and custody transfer concepts on the system performance of DTNs that deploy these features. For example, we are interested in understanding the performance difference when the DTN nodes move according to different mobility models e.g. random waypoint model or Zebranet-like model when DSR-like routing protocol is used with and without custody transfer feature. We are also interested in comparing the performance of a DTN that uses DSR-like routing protocol with custody transfer with the performance obtained when a 2-hop relay routing protocol [Princeton] without the custody transfer feature. Since the deployment of custody transfer incurs extra overhead, we are also interested in understanding when one should turn on this feature. In addition, we are also interested in understanding the additional benefits in deploying message ferries in a DTN when custody transfer feature is turned on. We conclude this report with some discussions of the future work that we intend to perform in the remaining time of the EDIFY project.

The rest of our report is organized as follows: we first give an overview of our enhanced DTN architecture in Section II. We describe the different network entities present in our architecture. We then describe the system operation, and the custody transfer feature in EDIFY. In Section 3, we first describe three routing approaches that have been proposed for forwarding intragroup bundles in DTNs and discuss their pros/cons. Then, we describe how intergroup routes can be discovered. In Section 4, we study the performance of intragroup bundle delivery. We study the impact of node densities and mobility models on intragroup bundle delivery ratio. In Section 5, we study the performance of intergroup communications in a DTN with a message ferry. We give some concluding remarks in Section 6 and also discuss additional work that we plan to do in the near future to substantiate further our preliminary recommendations.

## II. EDIFY DTN ARCHITECTURE

In the EDIFY architecture, there are several types of nodes, namely (a) regular DTN nodes, (b) the DTN name registrar (DNR), and (c) DTN gateway) Regular DTN nodes have the ability to send to, and receive bundles from other nearby nodes. Each DTN node may be configured with a default DNR from which each node acquires its name within the group. Every node in the DTN has one or more names – the DTN address within a group to which it belongs. An address comprises of two parts – a hierarchically-organized group name shared with all other members, and a group-specific name that is unique among all members.

In some DTN networks, some nodes take on additional roles and responsibilities. An example is the DNR nodes. A DNR node functions as the administrative node for a group and is responsible for communications with the parent group(s). A DNR offers the mandatory service of registering the members and visitors. It is responsible for ensuring the authenticity and eligibility of the nodes requesting to be registered. DNRs from different groups may form an overlay network. DTN gateways are DTN nodes that offer forwarding services to one or more destination groups. DTN nodes that perform forwarding services for different groups form the communication backbone within the DTN. A DTN gateway can advertise its services to the local registrar so that nodes in the group can identify which DTN gateway to use by asking the registrar or the DTN gateway can flood its local group with forwarding service advertisements. A message ferry is a special type of DTN gateway that moves around to allow communications between different groups that are not within transmission range of one another. Regular DTN nodes may form an ad hoc DTN network without the presence of a DNR. In an ad hoc DTN network, the DTN nodes behave similarly to the regular wireless ad hoc networks except that the DTN nodes support DTN functionalities like custody transfer which will be described in a later subsection.

### A. *System Operations in EDIFY*

Both infrastructure-based and infrastructureless networks are considered in EDIFY. Individual networks may be in wired or wireless domains. An infrastructureless DTN node needs to determine its location and neighbors upon power up. Thus, in a wired domain where only some nodes support DTN functionality, the DTN nodes can discover one another using an approach similar to a peer-to-peer network or ad hoc network [7]. They can send a neighbor discovery message with a TTL of one to a multicast address to which every DTN node will listen. If the new node does not hear any response, it sends another neighbor discovery message with increasing TTL until a sufficient number of responses are heard. To prevent too many simultaneous replies, each node should employ a random delay before replying. If no responses are found via multicast, a DTN node may attempt a broadcast in its own local subnet to see if it can discover any DTN nodes. In addition, a DTN node can attempt to contact any previously encountered DTN participants whose information is cached.

After the discovery phase, each node unicasts heartbeat messages periodically with its neighbors. The heartbeat message contains information about the node's identifier, the number of its own group members it can hear, the node's buffer availability, link duration/schedule (i.e., duration during which the node will be reachable), link characteristics (the number of hello messages received from neighbors), possibly the node's encounter histories (e.g., I have reached D before), and the number of external groups that it hears.

In an infrastructure-based network, e.g., a message ferrying system, special nodes that offer services to regular nodes exist. Such special nodes (referred to as message ferries) will announce their presence so that regular nodes can register with them to obtain services. Preliminary studies of message ferrying systems have indicated that they are very useful in delivering messages between partitioned networks [11][12][15]. Consider the example shown in Fig. 1 where there is a message ferry. The message ferry periodically broadcasts a ferry announcement message. Any nodes that wish to use the ferry's service should register with the ferry. The message ferry includes the currently registered groups in its ferry announcement messages so that nodes from one group can determine if they can reach nodes from another group via the message ferry. Note that a group may not be physically connected to another group (e.g., Network 1 and Network 4 in Figure 2) but the message ferry allows the two groups to communicate with one another via the store-and-forward mechanism.

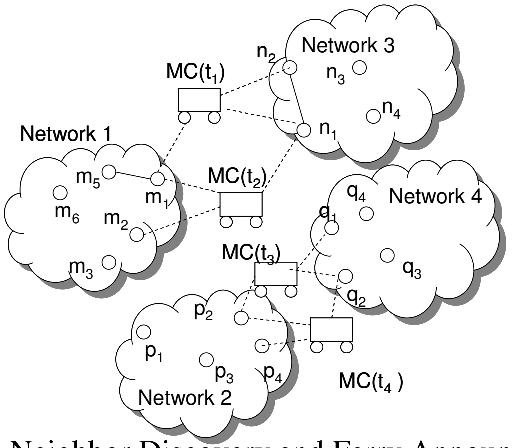


Figure 1. Neighbor Discovery and Ferry Announcement Procedures

### B. *Custody Transfer*

A custody transfer feature is proposed in [13],[17] to provide communications in a intermittently connected network. In their proposal, 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). Applications can optionally request the custody transfer feature on a per-message basis and they will receive a custody acknowledgement when their host system can find one or more nodes that are willing to take custody of the message. A node may agree to accept custody for messages initially and refuse to do so when its local node resources, e.g., buffers, become substantially consumed.

Messages may be fragmented or aggregated as they traverse the network [17]. Fragmentation may be performed at the source node if a contact does not have enough buffer space to accept the whole message. In addition, reactive fragmentation may happen when a receiving next-hop node is forced to fragment a message (message fragment) when a link become inoperative while a message is transferred. Reassembly of fragments into complete messages (or larger fragments) is possible when multiple fragments for the same destination together await an outgoing transmission opportunity. If reassembly of fragments does not occur in the network, it will be performed by the receiver. Potential problems that may occur with custody transfer are discussed in [17].

To study the impact of custody transfer feature on the packet delivery ratio, we implement this feature in a NS-2 simulator. The implemented custody transfer feature works as follows: when a DTN node has a message which it owns custodianship to send, it checks its cache to see if it has a route to the destination node. If it finds more than one route, it picks the one with the lowest cost (e.g., using hop count, delivery latency etc., as metrics). When a route is selected, it checks the DTN nodes included in this selected route to see which node is the best candidate for custody transfer, e.g., the closest DTN node that has buffer space available. Then, it sends a custodian request to that downstream DTN node. If the DTN node can accept the custodianship, it will respond with a custody acknowledgement. Otherwise, it sends a negative reply.

If the sending DTN node cannot find a route to the destination of the message, it will trigger its underlying ad hoc network layer to look for a route or neighboring nodes that are closer to the destination than itself. In addition, it will send a broadcast custody request message at the DTN layer. The DTN nodes that hear the custody request message and are closer to the destination than the issuer of such a request will send a custodian accept message to the sender of that request. At the ad hoc network routing layer, all DTN nodes that receive a route reply message with the DTN option flag set will set a bit in the appropriate position (according to its hop distance from the sending node of the route request) to indicate buffer availability before relaying the route reply message. That way, the sender knows whether or not that it can use that route. In a DTN environment, an end-to-end route may not exist. Thus, our dual-layer (at ad hoc network routing and DTN layers) approach allows us to identify downstream nodes to which we can forward the messages.

We did not implement the custody transfer complete message described in [17]. In our implementation, the sender transmits the bundle to another node if it does not receive custodian acknowledgement from the new custodian node in two seconds, and hence duplicated messages will be received at the destination node. Our custody transfer implementation avoids the head of line blocking problem described in [17] by allowing the DTN node to search through the queued bundles until it finds a bundle that can be sent to the next hop node.

## III. ROUTING SCHEMES FOR DTNs

### A. *Intragroup Routing Schemes*

Three categories of forwarding schemes have been proposed for DTNs. In the first category [20], the authors propose to use message ferries or data mules to gather data from stationary sources and deliver them to their destinations. However, for nodes that move, they can be message carriers themselves without having to resort to special message ferries. In the second category [21],[22], the authors propose using history-based routing where each node maintains a utility value for every other nodes 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 [22], the authors propose a probabilistic metric called delivery predictability at every node a for each known destination b. This metric indicates how likely it is that node a will be able to deliver a message to that 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 [22], 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 superior performance than epidemic routing [25]. We suspect that the transmission overhead (which is defined as the number of transmitted bytes over the number of generated bytes) for such schemes will be similar to that achieved using DSR-like routing protocol combined with custody transfer which we will elaborate more in the next paragraph. From Figure 5 in [22], it seems like the transmission overhead will be close to 38.6 for the case with 200 buffers.

In the third category [19], [23], 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 shorter deadlines. In this report, we use a DSR-like forwarding scheme together with custody transfer. We intend to simulate scenarios similar to those reported in [22], [19] so that we can compare the transmission overhead and the delivery ratios of these different schemes.

### B. *Intergroup Route Discovery*

In some network scenarios, one group of nodes (say Group 1) may not be able to hear another group of nodes (say Group 3) directly but they may hear members of a third group (say Group 2) that can communicate with Group 3 as shown in Figure 2. Different groups may use different algorithms to route packets within their own groups. We assume that different groups are willing to support a few common intergroup route messages to facilitate the ability for nodes from one group to route packets destined to another group. To minimize the need for all nodes to support inter-region routing, a gateway selection protocol [16] is used whereby only nodes which have been selected as gateways need to run an intergroup routing protocol.

Here, via an example with a message ferry, we describe how the nodes in a DTN environment can discover routes to other nodes. In Fig. 3, we have forty nodes that are partitioned into four isolated groups. There is a base station node in each group. The base station node is assumed to have a second long range radio that provides wireless backhaul link with higher bandwidth. To minimize the risk of potential enemy detection or energy consumption, the wireless backhaul links are only turned on periodically for short duration of time. In the example shown in Figure 3, we assume that BS1 (BS2) can communicate only occasionally with BS2 (BS3). Similarly, BS3 can communicate only occasionally with BS4.

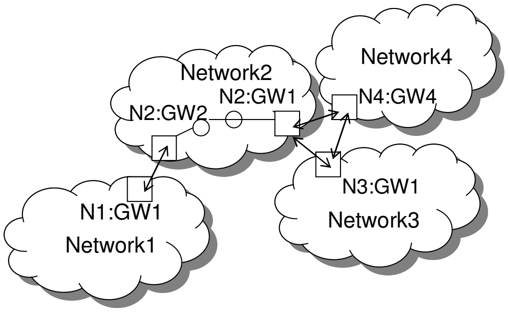


Figure 2. Intergroup Routing

Because these groups are isolated far away from one another, the groups can only communicate with one another either via the wireless backhaul links that are not always available or via a message ferry. We assume that message ferry broadcasts a service announcement message periodically as it moves along a fixed route. We also assume the service announcement message contains information on the groups that the message ferry can reach from previous trips. Other useful information like the estimated next visit time to those reachable groups may be included for more sophisticated forwarding decision.

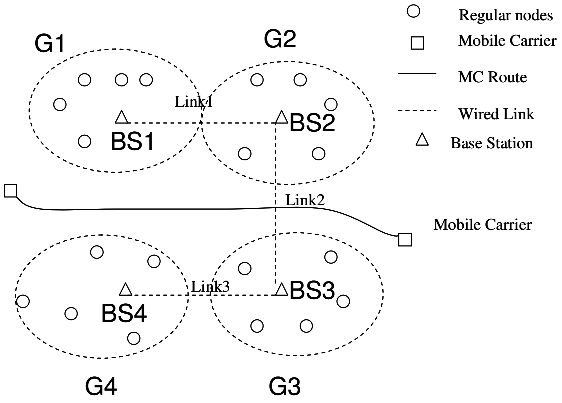


Figure 3. A 4-group DTN example

We assume that the intragroup routing protocol made use of the information provided by the underlying ad hoc routing protocol which is assumed to be DSR-like [7]. Whenever there is intergroup traffic, the nodes will evaluate to see if it consumes less cost (e.g., in terms of expected delivery delay) to send the traffic via the backhaul links or via the message ferry if both types of forwarding services are available. The base station will send announcements to inform the nodes whether or not it can provide intergroup forwarding services. For example, when Link 1 is not available, BS1 will inform all group members 1 that intergroup service is not available. Similarly, when Link 2 is not available, BS2 will inform BS1 that it cannot communicate with Group 3. Then, BS1 will inform Group 1 members that intergroup service to Group 3 is not available. Note that BS2 can delay such notification until its buffers are full or can notify BS1 immediately when Link 2 disappears.

Assume that Group 1 needs to communicate with Group 3; then a source node (which will be a node in Group 1) will send the traffic to BS1 and BS1 will forward it to BS2 when link 1 is available. If Link 2 is not available, the messages will be stored at the buffers at BS2 until Link 2 is available. Drop-from-front scheme can be used to replace old messages with new messages when the buffer at a base station is full. However, one can also use tail-drop scheme where no new messages will be accepted when the buffer at the base station is full. In addition, when different classes of messages are available, smarter buffer management schemes need to be designed to give different treatments to messages from different classes.

When the base station does not provide intergroup service or if the cost for sending such traffic using the backhaul route is higher than using a route via the message ferry, the regular nodes will use the service from the message ferry. Not all nodes can hear the message ferry. We assume that all nodes within a group that can hear the service announcement from a message ferry can provide forwarding services to/from the message ferry. Such nodes will be referred to as the gateway nodes. The gateway nodes can make periodic announcements to their group members that they can provide forwarding services. All regular nodes can cache information from such announcements but do not use the gateway nodes until they desire to use the service of the message ferry. The above approach is more proactive since the nodes within a group can find out where the gateway nodes are before they need to use the forwarding services from such nodes. Alternatively, the sending nodes can send gateway discovery messages to discover the gateway nodes. Once a sending node can identify the gateway node, it can then send intergroup traffic to the gateway node. This approach is more reactive and message delivery time may increase due to the need to perform gateway discovery.

## IV. PERFORMANCE STUDY OF INTRAGROUP COMMUNICATIONS

### A. *Impact of Node Densities in a DTN with Custody Transfer feature enabled*

In this section, we investigate how the presence of DTN nodes supporting custody transfer feature in a sparsely connected adhoc network impacts the system performance. DSR [18] is used as the default routing protocol. We conducted two sets of experiments. In our first set of experiments, we simulate a scenario where there are 40 nodes. The 40 nodes are distributed randomly in the following area (a) 1000x1000 $m^2$, (b) 1500x1500 $m^2$, and (c) 2000x200 $m^2$. First, we run some experiments assuming that the nodes do not support custody transfer i.e. they are just regular adhoc network nodes. Then, we run the same experiments assuming all nodes turn on the custody transfer feature. In our second set of experiments, we use the network illustrated in Figure 3 above and investigate the performance improvement when the custody transfer is invoked. In this subsection, we will discuss the results for the first set of experiment where we have 40 nodes randomly distributed in areas of different sizes. We defer the discussion for this second set of experiment to the subsection that discusses the benefits of using message ferrying.

In our first set of experiments, we use 10 source/destination pairs that each generates 1 packet/4 seconds. The source/destination pairs are randomly picked among the 40 nodes. The packet size is 512 bytes. The nodes move according to the random waypoint model with a maximum speed of 5 m/s. Table 1 shows the results we obtained without custody transfer and Table 2 shows the results we obtained when custody transfer feature is turned on. The metrics we use are:

(a) transmission overhead [19] which is defined as the number of transmitted bytes over the number of generated bytes. Note that in this case, the transmitted bytes include the routing overhead. Each routing message and each custody transfer request/acknowledgement message is assumed to be 35 bytes long.

(b) total number of control messages sent (including custody transfer acknowledgements)

(c) the average end-to-end delivery latency (denoted as Avg Delay in the tables)

(d) the packet delivery ratio (PDR),

(e) the average hop counts of the chosen path,

Table 1: Without Custody Transfer

| | Total # of control messages | Total # of data messages | Transmitted bytes over generated bytes | Avg delay | Delivery Ratio | Avg Hop count |
|---|---|---|---|---|---|---|
| 1000x1000 | 223391 | 6131 | 9.54 | 0.57 | 97.1% | 3.87 |
| 1500x1500 | 126035 | 7346 | 7.03 | 3.69 | 78.0% | 5.24 |

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| 2000x2000 | 52396 | 3870 | 3.27 | 6.54 | 48.0% | 4.64 |

Table 2: With Custody Transfer

|  | Total # of control messages | Total # of data messages | Transmitted bytes over generated bytes | Avg delay | Delivery Ratio | Avg Hop count |
|---|---|---|---|---|---|---|
| 1000x1000 | 229645 | 7376 | 10.49 | 1.59 | 100% | 3.9 |
| 1500x1500 | 158688 | 18091 | 13.16 | 43.4 | 99.9% | 5.7 |
| 2000x2000 | 113762 | 15098 | 10.39 | 259.4 | 98.6% | 5.25 |

From Table 1, we see that the packet delivery ratio starts to drop significantly (by 19%) when the node density changes from to $4x10^{-5}$ /m$^2$ to $1.8x10^{-5}$ /m$^2$ assuming that the transmission range is 250 m. The drop in PDR is 30% as the node density decreases from $1.8x10^{-5}$ /m$^2$ to $1.0x10^{-5}$ /m$^2$. Table 2 shows how the custody transfer feature significantly improves the packet delivery ratio for those scenarios where the node density is below $1.8x10^{-5}$ /m$^2$. In the 2000x2000 case, we see that the delivery ratio has dropped to 48% without custody transfer. However, the packet delivery ratio increases to 98.6% when the custody transfer feature is turned on. The additional price to pay for this improvement is an increase of transmission overhead by almost 300% and an increase of almost 100% in control overhead. We expect the delivery ratio to drop significantly when node density continues to drop and the custody transfer feature alone will not be enough to allow the sparsely connected nodes to communicate with one another. Thus, in EDIFY, we propose to use message ferries in very sparse adhoc networks. We will elaborate on the benefits of using message ferries in Section V.

### B. *Impact of different mobility models*

In this section, we describe an experiment we conducted to understand the impact of mobility models on the system performance. In this experiment, we simulate a scenario where 34 nodes are randomly distributed over an area of 1500mx1500m and the nodes move either according to random waypoint model or according to Zebranet movement [19]. For the random waypoint movement, the nodes have a maximum speed of 5m/s. For the Zebranet movement, we scale the node positions to be within 1500mx1500m area. To maintain similar node connectivities, we reduce the transmission range to 250 m (as compared to 1000 m in the original simulation reported in [19]). In the original Zebranet trace [19], the inter-sample interval is 8 minutes but we scale this interval to 8 seconds in the experiment we conducted. This means that in our experiment, the nodes move faster than those reported in the original Zebranet trace. Table 3 tabulates our results. The faster and more chaotic node movements that are based on the Zebranet trace result in higher average packet delivery latency (30.6 seconds compared to 22.6 seconds). The transmission overhead is also higher using the Zebranet mobility model as compared to that obtained using random waypoint mobility model.

Table 3(a): System Performance with ZebraNet Movement

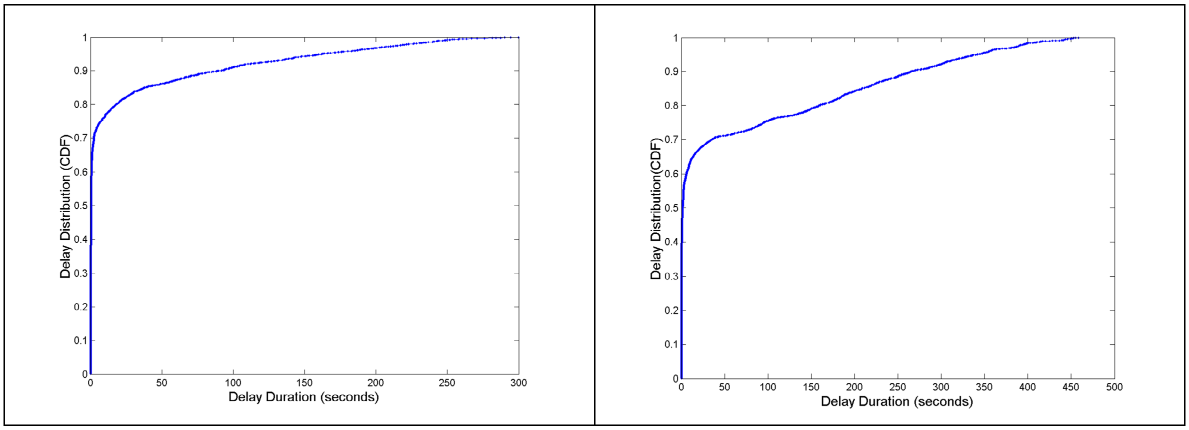
|  | Total # of control messages | Total # of data messages | Transmission Overhead | Avg delay | Delivery Ratio | Avg Hop count |
|---|---|---|---|---|---|---|
| RWP(5m/s) | 145594 | 13014 | 10.45 | 22.6 | 99.9% | 4.78 |
| Zebra | 129653 | 16521 | 11.23 | 30.6 | 99.9% | 4.56 |

We conducted another experiment where we scaled all the reported distances by 4 (to fit the locations to within the 1500mx1500m rather than the 6000mx6000m as described in [19]) and scaled the time unit such that the nodes will be moving at the same speed as reported in [19]). The result is tabulated below. We see that when the nodes move slower using the Zebranet mobility model, the average packet delivery latency and transmission overhead increases. The packet delivery latency increases from 30.6 seconds to 68 seconds while the transmission overhead increases from 11.2 to 15.1.

Table 3(b) System Performance with another Zebranet mobility model

| | Total # of control messages | Total # of data messages | Transmission Overhead | Avg delay | Delivery Ratio | Avg Hop count |
|---|---|---|---|---|---|---|
| With custody transfer | 169595 | 23301 | 15.14 | 68 | 99.9% | 5.98 |

The delay distributions obtained using the random waypoint model and using the Zebranet model are plotted in Figures 4(a) and 4(b) respectively. Here, we see that the Zebranet mobility model results in packet delivery latency that has a higher tail. The 90 percentile delay is 100 seconds using the random waypoint model but it is 260 seconds using the Zebranet model. If we translate the time unit from minutes to seconds in [19], the 90 percentile packet delivery latency achieved using the 2-hop relay forwarding scheme as reported in [19] is close to 1200 seconds. Thus, we suspect that the packet delivery latency will be smaller with the multihop routing approach as compared to the 2-hop relay approach. More simulation studies comparing these two forwarding schemes will be carried out in the near future.



(a) Using RWP (b) Using Zebranet model

Figure 4: Delay Distribution using different mobility models.

## V. INTERGROUP COMMUNICATION SCENARIO WITH MESSAGE FERRY

The scenarios in earlier section only involve a single group of nodes. In battlefields, sometimes different isolated groups need to communicate with one another. Traditional adhoc network routing protocols cannot be used for such scenarios. The 2-hop relaying protocol as described currently in [19] cannot be used for such communications as well. Using the network topology shown in Figure 3, we wish to investigate the performance that can be achieved for intergroup communications when wireless backhaul links and a message ferry are deployed. In this section, we conduct extensive simulation experiments to evaluate the impacts of the custody transfer feature, and the use of a message ferry on the message delivery ratio when the availabilities of the wireless backhaul links are varied both in terms of the relative on-off patterns and the percentages of their availabilities. We also explore the impacts of having limited buffers at the base stations and regular DTN nodes on the end-to-end message delivery ratio. We use ns-2 [8],[9] for our simulations. The common parameter values used in the simulation is tabulated in Table 4. Each group has ten nodes which are randomly distributed over an area of 1000m by 1000m. Thus, each group forms a sparse ad hoc network. All nodes support DTN functionalities. We assume that the regular DTN nodes communicate with one another via the 802.11 links with 2 Mbps link bandwidth, while the wireless backhaul link has a bandwidth of 5Mbps. We assume that only one type of message is used and that the message has a fixed size of 512 bytes. A complete message is sent using one bundle. We further assume that the message ferry has a buffer size of 400 messages. For each experiment, we measure the delivery ratio for the messages delivered via the wireless backhaul links and the message ferry separately. We also measure the contact time a message ferry has with a particular group during its route to help us understand the delivery ratio in each experiment. In addition, we also record the end-to-end message delivery times.

There are 10 pairs of traffic sessions where 4 pairs are single hop pairs (meaning requiring a traversal of only one backhaul link for delivery), 4 pairs are 2-hop pairs (meaning requiring a traversal of two backhaul links for delivery), and 2 pairs that are 3-hop pairs (meaning it needs to traverse 3 backhaul links) when the message ferry service is not available.

| **Parameter** | **Value** |
|---|---|
| Simulation Areas | 2000mx2000m |
| Group Size | 10 nodes/group |
| Wireless Link | 802.11(2M) |
| Wired Link | duplex link(5M) |
| Packet Size | 512bytes/packet |
| Traffic Pattern | CBR (interval:4sec/packet) |
| Buffer Size of Regular Nodes | Depending on experiments |
| Buffer Replacement Policy | Drop-from-front |
| Buffer of the Mobile Carrier | 400 messages |
| Speed of the Mobile Carrier | 15m/s |
| Traffic Load | 10 pairs |
| Simulation Time | 5000 seconds |

Table 4: Common Simulation Parameter Values

### A. *Impact of custody transfer on delivery ratio*

In our first set of experiments, we investigate how the custody transfer feature helps in the message delivery ratio. The backhaul link follows an on/off pattern shown in Fig 4 (Case 1) with a mean on/off cycle time of 100 seconds. We set the base station and the message ferry buffer size to be 400 messages. The buffer size for regular DTN nodes is set to 100 or 200 messages. We simulated four scenarios, namely, a) backhaul delivery without custody transfer, (b) backhaul delivery with custody transfer, (c) ferry delivery without custody transfer, and (d) ferry delivery with custody transfer. The availability of the backhaul links follows the Case 1 on/off pattern illustrated in Figure 5. The message delivery ratios achieved in these four scenarios are tabulated in Table 5. Our results show that the custody transfer feature improves the message delivery ratio significantly to near 90-92% with only the backhaul delivery mechanism and 89.4% (with 200 message buffers) with only the ferry delivery mechanism. The lower message delivery ratio for the message ferry case is due to the highly disruptive intragroup routes to the gateway nodes since each group is a sparse ad hoc network.

Table 5: Message Delivery Ratio for Experiment 1

| Buffer Size | 100 | 200 |
|---|---|---|
| Backhaul-delivery without custody-transfer | 55.6% | 58.4% |
| Backhaul-delivery with custody-transfer | 90.9% | 92.3% |
| Ferry-delivery without custody-transfer | 10.3% | 11.6% |
| Ferry-delivery with custody-transfer | 78.6% | 89.4% |

### B. *Impact of Link Pattern on End-to-end Message Delivery*

In this experiment, we fix the link availability to be 20% but vary the link availability patterns of the 3 links to investigate its impacts on the message delivery ratio. The 3 link patterns we use are shown in Figure 5. The on/off times follow an exponential distribution with a certain mean on/off times to mimic link patterns shown. In Case 1 and Case 2, each link is available for an average of 20 seconds and not available for an average of 80 seconds. The two cases only differ in the relative positions of the link availabilities. To achieve the on/off patterns shown as Case 1 in Figure 6, we generate a random on time for link 1 (say x1), then schedule for link 2 to be on only after time t1 and it will be on for another random on time (say x2), etc. In Case 3, each link is available for an average of 30 sec and not available for an average of 120 sec. So, the link availability is also 20%.

In our second experiment, we set the mean on/off period to be 200 seconds. The custody transfer feature is turned on. The DTN nodes only use the backhaul links (no message ferry service is provided). The results are tabulated in Table 6. The number expressed in seconds is the mean end-to-end message delivery time. There are a few interesting observations we can make from these results. First, the delivery ratio for the traffic session between Group 1 and Group 3 (indicated by 1-3 pair) is lower than other pairs because the route between the source node and the base station breaks more frequently than other traffic pairs, e.g., Group 3 to Group 1. The mean end-to-end delivery time depends on various factors, e.g., the link availability pattern, the connectivity between the source node and the base station, etc. For example, even though session 1-3 and session 3-1 are both 2-hop pairs, the average end-to-end message delivery time for session 3-1 is higher than that achieved for session 1-3 using Case 1 link patterns because the traffic from Group 3 requires at least two on/off cycles to reach Group 1 but the traffic from Group 1 to Group 3 only needs one on/off cycle. In general, the mean end-to-end message delivery times are higher for traffic sessions that traverse more backhaul hops. However, there may be situations where this is not true. For example, the mean end-to-end delivery time for traffic session 1-4 is smaller than the mean end-to-end delivery time for traffic session 1-3. This is due to the fact that the source node for this 1-3 pair is sparsely connected to the base station in Group 1 while the source node for the 1-4 pair has a very reliable route to the base station in Group 1.

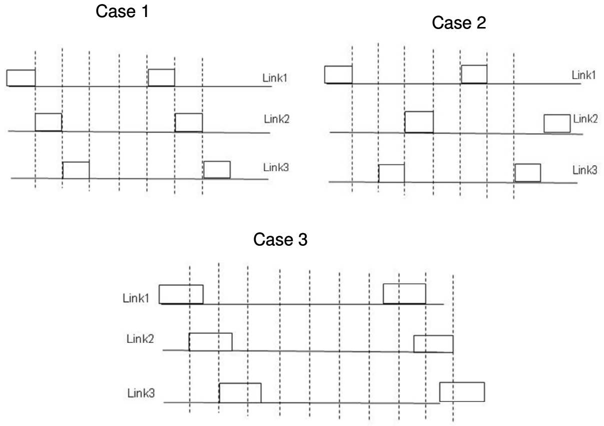


Figure 5. Various Link Availability Patterns

Table 6: Message Delivery Ratio and Mean End-to-end Message Delivery Time for Experiment 2.

| | Link Pattern 1 | Link Pattern 2 | Link Pattern 3 |
|---|---|---|---|
| 1-3 pair | 78.7%<br>277.2seconds | 82.0%<br>410.8second | 75.1%<br>385.9seconds |
| 3-1 pair | 86.7%<br>352.5seconds | 84.7%<br>337.1seconds | 82.1%<br>306.4seconds |
| 2-4 pair | 96.2%<br>131.4seconds | 97.0%<br>278.3seconds | 99.1%<br>144.5seconds |
| 4-2 | 96.0% | 97.0% | 96.9% |

| | | | |
|---|---|---|---|
| pair | 229.3seconds | 145.9seconds | 166.9seconds |
| 1-4 pair | 75.9%<br>188.1seconds | 77.4%<br>432.0seconds | 79.3%<br>239.3seconds |
| 4-1 pair | 85.0%<br>444.6seconds | 96.8%<br>240.9seconds | 72.2%<br>345.7seconds |
| 1-hop pairs | 88.7%<br>138.5seconds | 89.4%<br>148.3seconds | 87.9%<br>193.7seconds |

### C. *Buffer Size Study*

In the third experiment, the nodes use either (a) only the backhaul links, (b) only the message ferry, or (c) both backhaul links and message ferry to deliver intergroup traffic. We set the message ferry and base station buffer size to be 400 messages each. We then vary the buffer size of the regular DTN nodes to see what its impact on the message delivery ratio and end-to-end message delivery times. A mean on/off cycle of 200 seconds is used for this third experiment. The results for the message delivery ratio, the end-to-end message delivery time (denoted as delay in Figure 6) and the overhead of control messages sent are plotted in Figures 6, 7, and 8. The results indicate that a delivery ratio of 90% is achievable even with 20% link availability.

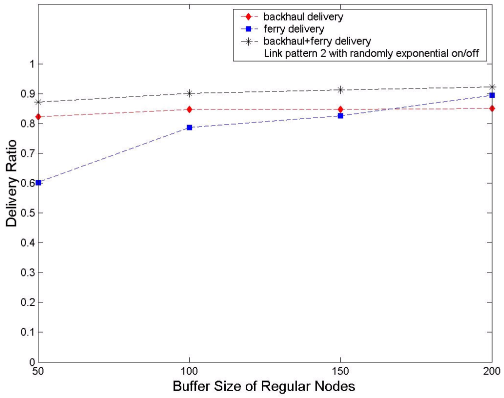


Figure 6: Message Delivery Ratio

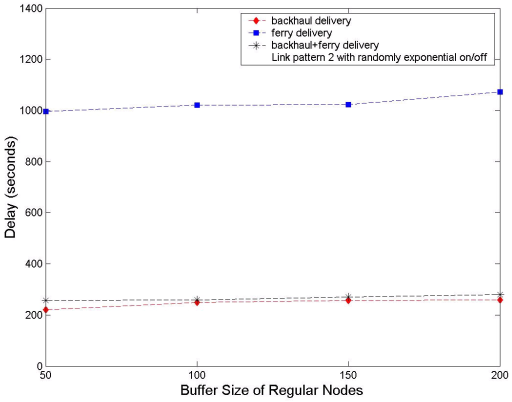


Figure 7: End-to-end Message Delivery Time

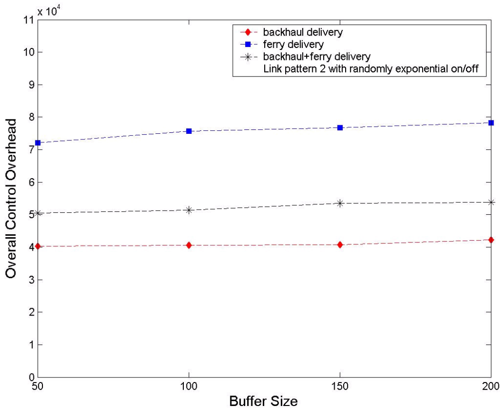


Figure 8: Control Overhead

## VI. CONCLUDING REMARKS/FUTURE WORKS

In this report, we illustrate the usefulness of the custody transfer feature in an intragroup communication scenario and how the transmission overhead and achievable packet delivery latency varies in different mobility models. We also caution that the custody transfer feature alone does not guarantee communications in a very sparsely connected network. We propose to use message ferries for scenarios where the nodes are very widely distributed. Via an example, we show that both the message ferry and the custody transfer feature can improve the end-to-end message delivery ratio in a multihop scenario where link availability can be as low as 20%. In particular, our results indicate that one can achieve a delivery ratio as high as 90-99% with appropriate buffer allocations. We also provide some preliminary insights on the design factors that influence the end to end delivery ratio in the widely disconnected scenario, e.g., the link availability patterns and buffer allocation strategies. In addition, we assume in this report that the traffic demands from one group to another do not vary with time and that the link availability follows exponential on/off distribution. In real world scenarios, the traffic demands and the link availabilities may be changing dynamically so one may not be able to predict the maximum required buffer size for the base station. So, some work needs to be done to estimate the maximum required buffer size given some rough estimates of the nodes' velocities, and mobility models in different environments.

The results reported here are work-in-progress. There are several important tasks that we intend to explore before making a final recommendation:

1. In our current simulations, we only assume CBR/UDP type of traffic. In real battlefields, there are different types of traffics e.g. in [25], [26], the authors report that there are at least 4 types of traffic, namely (a) high rate report with source rate of 40 Kbps (1Kbytes packet), (b) low rate report with source rate of 0.8Kbps (100 byte packet), (c) high rate traffic which is either 120Kbps or 200 Kbps with 1 Kbytes packet, (d) bidirectional traffic with source rate 40 Kbps. Different number of flows of each type may exist at any particular time. We intend to simulate similar traffic mixtures to see if the preliminary conclusions we have still hold in scenarios where there are different traffic mixtures. We also intend to explore applying different buffer management policies and service disciplines to different types of traffic.
2. Most of our simulation results are based on random waypoint mobility model even though we have some preliminary results using the mobility model derived from the Zebranet trace. We intend to include some results based on the mobility model obtained from the FCS data.
3. There are several tuning parameters in the different forwarding schemes e.g. we have the cache timeout value in the DSR scheme, the aging constant in the probabilistic routing scheme. It will be

useful to report on how sensitive the system performance is to the values of the tuning parameters and the comparisons of the different schemes using their best parameter values.

4. Messages may be fragmented into different bundles. In this report, we have not investigated such scenarios. We hope to explore this issue in the coming months.
5. We intend to explore more intergroup communication scenarios so that we can understand better how the intergroup forwarding scheme should be designed.

## REFERENCES

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

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

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

[4] 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.

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

[6] M. Chuah, L. Cheng, and B. D. Davison, “Enhanced Disruption and Fault Tolerant Network Architecture for Bundle Delivery (EDIFY)”, to appear in Globecom, 2005

[7] C. E. Perkins and E. M. Royer, ”Ad-hoc On-Demand Distance Vector Routing”, *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*, pp 90-100, Feb 1999.

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

[9] J. H. Saltzer, D. P. Reed, and D. D. Clark, “End-to-End Arguments in System Design”, *ACM Transactions on Computer Systems*, 2(4):277-288, November 1984

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

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

[12] 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.

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

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

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

[16] W. Ma and M. Chuah, “Comparisons of Interdomain Routing Schemes for Heterogeneous Adhoc Networks”, Proceedings of WoWMoM, 2005.

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

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

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

[20] R. Shah etc, “Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks”, Elsevier Ad Hoc Networks Journal, Vol 1, pp 215-233, Sept, 2003

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

[22] 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.

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

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

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

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