Performance Evaluation of an Encountered Based Multicast Scheme for Disruption Tolerant Networks

Y. Xi, M. Chuah
Department of Computer Science & Engineering
Lehigh University
yox205@lehigh.edu, chuah@cse.lehigh.edu

***Abstract*—** Some ad hoc network scenarios are characterized by frequent partitions and intermittent connectivity. Hence, existing adhoc routing schemes which assume the existence of end-to-end paths do not work in such challenging networks. A store-and-forward network architecture known as the disruption tolerant network (DTN) has been designed for such challenging network environments. Several unicast and multicast routing schemes have been designed for DTNs. However, the existing multicast routing schemes assume a route discovery process that is similar to the existing adhoc network routing approach, and hence will not work well in very sparse network scenarios. Thus, in this paper, we explore an encounter-based multicast routing (EBMR) scheme for DTNs. Our scheme uses fewer hops for message delivery. We present an analytical framework for estimating the delivery performance of the EBMR scheme, and present some analytical and simulation results to show that the EBMR scheme can achieve higher delivery ratio while maintaining high data transmission efficiency compared to other multicast strategies.

***Index Terms*—disruption tolerant networking, multicast routing, redundancy.**

I. INTRODUCTION

With the advancement of technology, we can find many wireless devices e.g. sensors, PDAs, laptops etc. Such devices can form wireless ad hoc networks dynamically without any infrastructure support. Much work has been done in the past to design unicast routing schemes for ad hoc networks [1]. However, the unicast routing schemes designed for ad hoc networks often assume that that an end-to-end path exists between a source/destination pair and hence are not suitable for challenging network environments where the nodes experience intermittent connectivity and frequent partitions. In addition, much design has also been done for delivering multicast traffic in ad hoc networks e.g. [3],[4]. Again, such multicast routing schemes often assume a multicast tree can be maintained for delivering multicast traffic. It is difficult to maintain a multicast delivery tree in challenging network environments with frequent partitions and intermittent connectivity among the nodes.

Recently, a new network architecture [5] called the Disruption Tolerant Network (DTN) has been proposed to allow partitioned nodes or clusters of nodes to communicate with one another. Recent research interests in this area include network architecture design [5], and different routing algorithms for DTNs [7][8],[9],[10]. Most of the routing schemes designed are delivering unicast traffic. However, many potential DTN applications require efficient network support for multicasting. For example, in a battlefield, soldiers in a platoon need to share information about their surrounding environment among one another. In a disaster rescue operation, it is critical to share information about victims and potential hazards among the rescue workers. Although group communication can be implemented by sending a separate copy of data via unicast to each user, this approach is not efficient and can potentially consume much power in small mobile devices that the soldiers or rescue workers carry. Thus, efficient multicast services are essential to support such applications.

Several DTN multicast routing schemes have been designed for DTNs e.g. [12],[13]. These DTN multicast routing schemes rely on a route discovery process that is similar to traditional adhoc routing schemes and hence may not be able to perform well when the network becomes very sparse. In [14], the authors overcome this problem by allowing nodes in very sparse environment to use high power transmission to complete the route discovery process and use message ferrying to deliver data packets. Even though such an adaptive scheme can significantly improve the delivery performance, this approach assumes that mobility of the nodes can be controlled which may be infeasible in some network scenarios. Thus, in this work, we design an encounter-based multicast routing scheme that does not assume controlled mobility of nodes. Our contributions in this paper are: (i) we design an encounter-based multicast routing scheme that provides high delivery performance (high delivery ratio and data efficiency), (ii) we provide an approximate analysis for the average number of hops taken and the average per-hop delay for our scheme, (iii) via simulations, we compare our scheme with other schemes.

The remainder of this paper is organized as follows. We provide a brief review of related work in Section 2. In Section 3 we describe several multicast strategies we consider in this work including the new encounter-based multicast routing scheme that we design. In Section 4, we present an analytical framework for estimating the delivery performance of the EBMR scheme. Then, we show our analytical framework is useful in estimating the delivery performance in sparsely connected ad hoc network scenarios. In Section 5, we describe our simulation setup and present some simulation results We conclude in Section 6 with some discussions on future work.

II. RELATED WORK

A. *Routing in Intermittently Connected Networks*

Several routing schemes have been proposed for DTNs [7],[8],[9],[10]. These different schemes can be grouped into three categories. The first category [7] uses special nodes called ferries to deliver messages between partitioned networks. Ferry routes have significant effect on the data delivery performance, hence they need to be designed efficiently. The second category [8],[9] uses multihop routing approach where contact history information is used to determine the next hop node to pass a message. For example, in [9], a probabilistic metric called delivery predictability is used to determine if a node needs to pass any stored messages to a new contact that it comes across. More discussions on this scheme will be provided in Section III. The third category [10] uses a two-hop routing approach where the intermediate nodes that receive messages from any source have to store the messages until they can deliver the messages when they come into contact with the destinations of the messages. Sometimes, erasure-coding is used to encode and divide the message into multiple blocks and these different blocks are sent to different relays to increase the chances of a destination receiving a particular message since the destination only needs to receive a certain fraction of the encoded blocks to reconstruct the original message.

B. *Multicast Routing Schemes in DTNs*

Several multicast routing schemes have been designed for DTNs, namely (a) DTBR [12], (b) OS-Multicast [13], and (c) Context-Aware Multicast Routing (CAMR) [14]. DTBR is a tree-based multicasting algorithm. DTBR assumes that each source node of the multicast group has complete knowledge or a summary of the link states in the network. During the lifetime of a multicast session, DTBR requires an upstream node to assign the receiver list for its downstream nodes based on its knowledge of the current network topology. The downstream nodes are allowed to forward bundles only to the receivers in the list. The custody transfer feature is enabled for those DTN nodes along the multicast tree. However, since the network topology changes frequently, it is not easy to maintain the multicast delivery tree. In addition, the receiver list cannot be adjusted by intermediate nodes once it is decided by upstream nodes, which means newly discovered delivery opportunities cannot be used by intermediate nodes.

Due to the limitations of DTBR, OS-multicast [13] was proposed. OS-multicast relies on a DSR-like routing to build a knowledge base of the link state and network topology. Unlike DTBR, OS-multicast let each intermediate node maintain a tree rooted at itself to all the receivers and adjust the receiver list according to local knowledge of the network topology. Via simulations, the authors [13] show that OS-multicast achieves good performance when the probability of the link unavailability is high. However, all simulations are based on a network of 25 nodes deployed in a 1000×1000 m^2 area, which is still quite well-connected. The authors in [14] show that the performance of OS-multicast degrades when the network becomes sparser. Moreover, OS-multicast relies on a DSR-like route discovery process to build a knowledge base of the current network topology. Such a process will not work in a very sparse network environment.

In [14], the authors propose a context-aware multicast routing (CAMR) scheme where nodes are allowed to use high power transmissions when the locally observed node density drops below a certain threshold. Each node maintains a 2-hop neighborhood information, and hence can deliver traffic without invoking a route discovery process if all receivers are within its two-hop neighborhood. In addition, the nodes can act as message ferries when they discover they are in a very sparse neighborhood. The combined high-power route discovery process and the message ferrying features allow CAMR to achieve much higher multicast delivery ratio than DTBR and OS-multicast schemes. However, CAMR still relies on a route discovery process that is similar to the traditional adhoc routing approach, and also relies on the ability to control node movements. Thus, in this work, we are motivated to design a multicast routing scheme that is purely based on node encounters.

III. DTN MULTICAST STRATEGIES

In this section, we describe a few of the DTN multicast strategies that are considered in this work:

A. Unicast-based Multicast Strategy

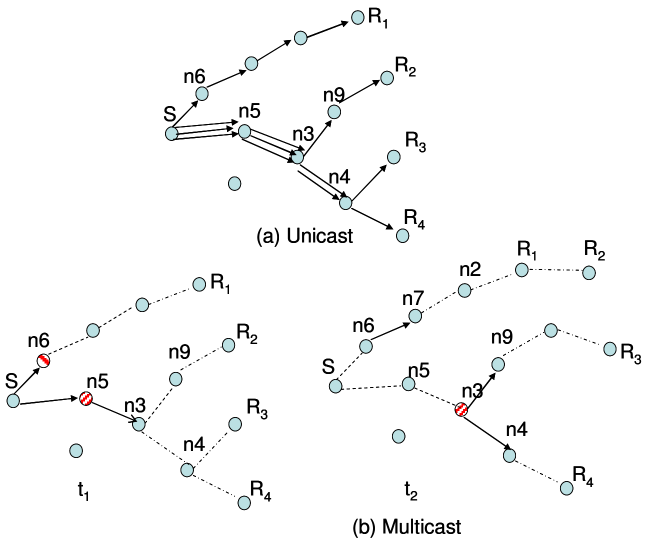


Figure 1: Unicast and Early Decision-Based Multicast Strategies

For Unicast (shown in Figure 1(a)), the source duplicates one message per receiver, and delivers it using the shortest route to each receiver. The diagram indicates that an intermediate node may receive multiple copies of the same message. Any DTN unicast routing scheme discussed in Section II can be used to delivery such unicast messages. In this work, we use the original Prophet scheme [9] as the DTN routing scheme for the Unicast approach.

B. Early Decision-based Multicast Strategy

In this strategy, whenever a node, n_1 , with a multicast packet to deliver, encounters another node, n_2, it will check to see if that node has a higher delivery predictability value than itself to any receivers of that multicast packet. If the value from n_2 is higher, then a duplicate copy of the multicast packet will be sent to n_2. n_1 will keep a copy of that multicast packet until it has found a next-hop node for all the receivers of that multicast packet. Thus, multicast messages are duplicated at every branching node based on how many downstream neighbors the node has in the multicast tree. For example in Figure 1(b), at time t_1, S encounters two nodes, n5 and n6, simultaneously. Based on the delivery predictability values S hears from n5 and n6, S determines that n5 has higher delivery predictability values to R_2 , R_3, and R_4 than itself. Similarly, S finds that n6 has higher delivery predictability value to R_1 than itself . Thus, S duplicates the multicast packet, sends one to n6 with a header that includes the receiver R_1, and send another with a header that includes receivers R_2, R_3 , and R_4 to n5. n5 encounters n3, and finds that n3's delivery predictability to R_2, R_3, and R_4 is higher than itself, so n5 forwards its multicast copy to n3. At time t_2, n3 encounters nodes n4, and n9. n4 has higher delivery predictability values to R_4, and n9 has higher delivery predictability value to R_3. Thus, n_3 duplicates the packet, sends one with a header that includes an identifier for R_3 to n9, and sends another with a header that includes an identifier for R_4 to n4. Since the header of the multicast packet which n3 receives includes R_2, n3 picks a node it encounters to send a multicast packet for R_2 when WT expires. Note that because an early decision is made to split the receiver list, the multicast packet which n2 receives from n7 only contains a header for R_1 but not R_2. Even though n2 may have higher delivery predictability value to R_2 at time t2, n2 cannot send this packet to R_2. Thus, with this strategy, some multicast packets may not reach the receivers due to the early decision to split the receiver list in the header.

C. Encounter-based Multicast Routing (EBMR) Scheme

In EBMR, each node broadcasts (within 1 hop) a beacon periodically. Each node, say, A, also establishes a probabilistic metric called delivery predictability for each known destination B. This metric indicates how likely it is that node A can deliver a message to that destination. The delivery predictability values from this node to all other nodes are included in the beacon messages. Such delivery predictability values are updated upon receiving beacons from other nodes. The delivery predictability value 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. The three equations used for updating the delivery predictability are the same as in Prophet [9]:

$$P(a,b) = P(a,b)_{old} + (1 - P(a,b)_{old}) * \alpha \text{ - Eqn 1(a)}$$

$$P(a,b) = P(a,b)_{old} \times \gamma^k \qquad \text{- Eqn 1(b)}$$

$$P(a,c) = P(a,c)_{old} + (1 - P(a,c)_{old}) * P(a,b) * P(b,c) * \beta \text{ - Eqn1(c)}$$

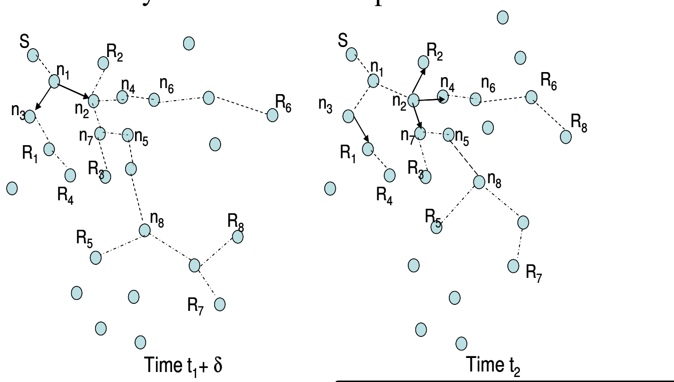
where P(a,b) denotes the delivery predictability of reaching node b from node a and α is an initialization constant chosen from the range [0,1].

Eqn 1(a) allows a node to update the metric whenever a node is encountered so that nodes that are often encountered have a high delivery predictability. If a pair of nodes does not encounter each other for a while, they are less likely to be good forwarders of messages to each other, thus the delivery repdictability values must age. The aging equation is shown in Eqn 1(b) where γ is the aging constant and k is the number of time units that have elapsed since the last time the metric was aged. The delivery predictability also has a transitive property that is based on the observation that if node a frequently encounters node b, node b frequently encounters node c, then node c probably is a good node to forward messages destined to node a. Eqn 1(c) shows how this transitivity affects the delivery predictability where β is a scaling constant that decides how large impact the transitivity should have on the delivery predictability. We use the same values in [9] for α, β, and γ : α is set to 0.75, β is set to 0.25 and γ is set to 0.98.

Each node maintains an N*N matrix (where N is the total number of nodes in the system) where each row i records the delivery predictability of node i to the other (N-1) nodes (the diagonal entry (i,i) is not used). Every time a node's beacon timer expires, that node will use Eqn 1(b) to update the delivery predictability values of those nodes that it has lost contacts with. If a node, n_1, hears another node (say n_2)'s beacon which contains the delivery predictability values from that node to other (N-1) nodes, then n_1 uses those values it hears from n_2, Eqns 1(a), and 1(c) to update its own delivery predictability values to other nodes. Note that instead of using the same equations as Prophet to compute delivery predictability, we can use other approaches e.g. [8] for delivery cost calculations in EBMR.

Since in the past, we have discovered that the Prophet scheme often uses large number of hops to deliver unicast messages, we introduce two new features in EBMR to ensure that the multicast messages can be delivered using fewer hops. Our enhancements should improve the delivery performance of unicast messages too. In EBMR, each node will not pass any bundle to another node that it encounters unless that node has delivery predictability higher than a delivery threshold (P_{thresh}) or a wait timer (*WT*) expires. The wait timer prevents the messages from being held up for too long at the source or any intermediate node. When a node using the EBMR scheme receives a multicast bundle, it will pick as many nodes as needed with the highest delivery predictability (that exceeds P_{thresh}) to each of the multicast receivers. The node will cache the data if no such next-hop node is found until a wait timer *WT* expires. If *WT* expires, then the node will simply pick a node with the highest delivery predictability to a multicast receiver whose next-hop node has not yet been selected. These two enhancements improve the delivery performance. Every relay node (including the source node) can have the opportunity to select a next-hop node with delivery predictability higher than P_{thresh} within the *WT* time. This results in fewer numbers of hops being taken to deliver messages to all multicast receivers. Thus, the overall delivery ratio and data efficiency of EBMR should be better than the early decision-based multicast strategy.

In Figure 2, we show a source S which sends a multicast packet to eight receivers, R_1 to R_8 respectively. In Figure 2, S encounters n_1 at time t_1 so S sends n_1 a multicast packet with a header that includes the identifiers for all 8 receivers. n_1 encounters nodes n2 and n3 at time ($t_1+ \delta$). n_1 looks at the rows corresponding to n_2 and n_3 in its NxN delivery predictability matrix, and see which node has the highest delivery predictability value for a particular receiver. The predictability value to any receiver has to be higher than P_{thresh} (which is set to say 0.1) before that node can be considered as a suitable next-hop node. Using the table shown in Figure 3, one can see that node n_2 should be used to reach receivers R_2 , R_3 , and R_5 while node n_3 should be used to reach receivers R_1 and R_4 . Thus, n_1 will create two multicast packets: one with a header which includes the identifiers for R_2 , R_3 , and R_5 ; the other with a header which includes identifiers for R_1 and R_4. Node n_1 records the information that no suitable next hop node has yet been found for receivers R_6, R_7, R_8 at time t_1+δ. At time t_2 , let us assume that the nodes have moved such that the delivery predictabilities for R_6 , R_7 via n_2 have increased above P_{thresh} . Then, node n_1 will create yet another multicast packet with a header that includes the identifiers for R_6 and R_7. Note that at time t_2 , node n_1 still has not found a suitable next hop node for R_8. When WT expires, node n_1 will select a next-hop node with the highest delivery predictability to R_8 (even if this value does not exceed P_{thresh}), and forwards a copy of the multicast packet to that node. Node n_1 only removes this multicast packet from its buffers after it has successfully selected a next hop node for all receivers.



Nxt Hop	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8
n_2	0.001	0.33	0.2	0.001	0.1	0.012	0.002	0.002
n_3	0.25	0.001	0.001	0.15	0	0.001	0	0

Nxt Hop	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8
n_2	0.001	0.33	0.2	0.001	0.15	0.11	0.11	0.05
n_3	0.25	0.001	0.001	0.15	0	0.001	0	0

Figure 2: Encounter Based Multicast Routing (EBMR) Scheme

Note that if one uses the early decision-based multicast (EDM) strategy described in Section III.B, then, every intermediate node that receives a multicast packet will select the best next-hop node for all the receivers using its NxN matrix, and will duplicate as many packets as needed to be sent to the selected next-hop nodes for all receivers. In Figure 3 example, if EDB strategy is used, then at time t_1 , node n_1 generates two multicast packets: one with a header that includes identifiers for R_2 , R_3 , R_5 , R_6 , R_7 , and R_8, and the other with a header that includes identifiers for R_1 and R_4. Since the delivery predictability for R_6, R_7 and R_8 from n_2 is low at time $t_1+\delta$, such early decision may result in the duplicated multicast packets taking longer paths to reach these 3 receivers.

D. *EBMR with replication*

Replication forwarding has been shown [8] to improve DTN message delivery performance in unicast scenarios. Thus, an additional enhancement one can use with EBMR is to allow a source node to select *K* next-hop nodes for each multicast receiver. In Figure 3, we show how EBMR with replication strategy works. Only the source is allowed to select K (K=2 in Figure 3) next-hop nodes for each multicast receiver. Let us assume that the source S wants to send a multicast packet to the four receivers R_1 to R_4. At time t_1 , based on the NxN delivery predictability matrix S maintains, S selects nodes n_1 and n_2 as the next hop nodes for the first copy of a multicast packet. At time t_2, nodes n_1 ,and n_2 forwards the multicast packet they received to nodes n_4, and n_6 respectively. Meanwhile, S has moved to a new location and determined that it can reach R_4 directly, and that n_3 can be used to reach R_2. As S has so far selected only one next hop node n_1 (n_2) for R_2 (R_4) respectively, therefore S can still send the multicast packet directly to R_4, and send another copy to n_3 to be delivered to R_2 since K=2. It is obvious from Figure 4 that using replication with EBMR scheme allows us to have shorter delivery latency at the expense of more transmissions.

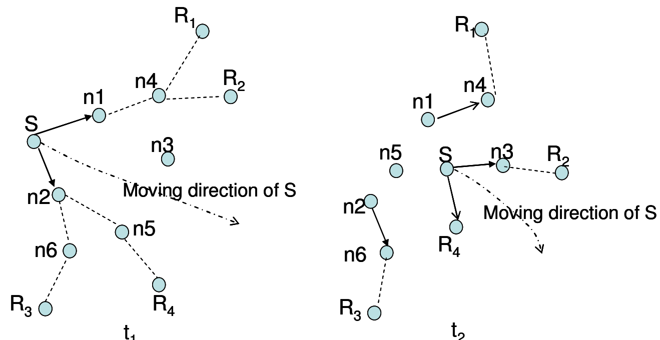


Figure 3: EBMR with Replication (M-K=2)

IV. ANALYSIS OF THE EBMR SCHEME

In this section, we present an analytical framework to estimate the delivery performance of the EBMR scheme. This analytical framework allows us to investigate how the choices of the two tunable parameters in EBMR namely P_{thresh} and *WT*, affect the delivery performance. Since we are interested only in the sparsely connected ad hoc network scenarios, our analysis assumes that the node density is low e.g with an average number of neighbors that is below 1.5. We first present an analysis for determining the average number of hops, *Lm*, taken to reach a receiver with different P_{thresh} and *WT* values. Then, we present an approximate analysis for the average one-hop delay, d_{hop} . Such analysis allows us to estimate the average end-to-end delay. Next, we present simulation results to show that the approximate analysis we have is close to the observed simulation results. We have also studied the sensitivity of the delivery performance to different values of the two tunable parameters.

A. Analysis for *Lm*

Given P_{thresh}, *WT*, and other network parameters (like, *N x N, M,* v_{ave}, *R,* etc), we estimate the average number of hops, Lm, when EBMR is used.

Table 1: Notation

N x N	Network area
M	Number of nodes
v_{ave}	Average speed of node movement
R	Transmission range

P_{thresh}	Delivery Probability Threshold
WT	Wait Timer in EBMR

1. Determining p_i ,the delivery probability towards the destination after one hop

In EBMR (P_{thresh}, WT), a node holding a packet may have two delivery choices:

a) Before *WT* expires, it may send the packet out when encountering a node with a delivery probability to the destination higher than P_{thresh} .

b) When WT expires, it will simply pick up a node with the highest delivery probability to the destination.

Suppose during WT, a node will encounter n_e nodes. n_e can be calculated below [11]:

$$n_e = 2*R * v_{ave} * WT * d \quad (2)$$

where, d is node density. Then, denote:

p_0 : the probability that none of these $n_{e|WT}$ encounters has a delivery probability that exceeds P_{thresh} . In other words, a node has to hold the packet until W*T* expires.

p_i : the average delivery probability towards destination after one hop.

p_e : the new delivery probability towards destination if a packet is relayed before WT expires.

p_d : the new delivery probability towards destination if the packet is relayed at the expiration of WT.

Therefore,

$$p_i = (1- p_0)* p_e + p_0 * p_d \quad (3)$$

2. Determining L_m

Next, we use a similar argument as in [2] to derive the average hop L_m. We denote *lm* as the number of hops a packet takes to reach its destination, and let P(*lm* > n) be the probability that the packet has not reached the destination after n hops. We then have,

$$P(lm > n) = \prod_{i=1}^{n}(1-p_i) \quad (4)$$

Since each hop is independent of one another, we have

$$P(lm > n) = (1-p)^n \text{ , where } p_i = p \quad (5)$$

Eqn (5) indicates that *lm* is geometrically distributed and hence $E[lm]= L_m = 1/p$.

3. Derivation of p_0, p_e, p_d .

(a) determining p_0

p_0 can be derived as follows:

$$p_0 = \prod_{i=1}^{n_{e|WT}} p_{0|i} \quad (6)$$

where,

(i) $p_{0|i}$ means the probability that the i^{th} encounter does not have delivery probability higher than P_{thresh}. In other words, the packet will not be sent to the i^{th} encounter.

(ii) $n_{e|WT}$ is the number of encounters during *WT*.

To understand how to derive p_0, consider the picture in Figure 4. Let t_a be the time taken for the delivery probability to a destination to drop from 1 to P_{thresh} when a node no longer can hear any nodes that can reach the destination. Let us assume that the source node encounters node a1 at time t_1 , nodes b1, b2 were encountered by node a1 during [t_1 - t_a, t_1], nodes c1, c2 were encountered by node b1 during [t_2- t_a, t_2] where $t_2 \in [t_1 - t_a, t_1]$ is the time node a1 encounters node b1 , and nodes c3, c4 were encountered by node b2 during [t_3- t_a, t_3] where $t_3 \in [t_1 - t_a, t_1]$ is the time node a1 encounters node b2. Then, given P_{thresh}, node S will not relay the packet to node a1 as long as none of these nodes b1,b2,c1,c2,c3, and c4 encounter any receiver during the indicated periods.

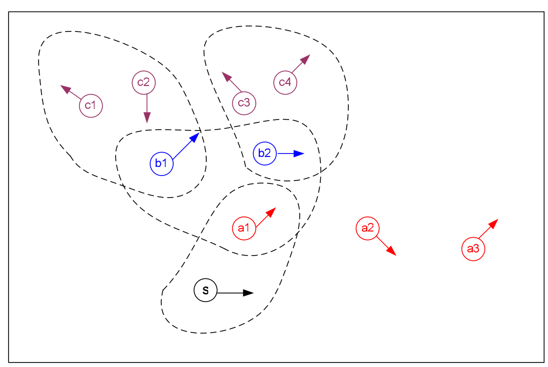


Figure 4: Node Encounters

Therefore, we have,

$$p_{0|i} = \frac{\binom{M-2-(i-1)}{1}}{\binom{M-1-(i-1)}{1}} \prod_{j=1}^{j=(n_{e|t_a+T_c})} \left(\frac{\binom{M-2}{n_{e|t_a+T_c}-1}}{\binom{M-1}{n_{e|t_a+T_c}-1}} \right)^j$$

where, *i=1,…,* $n_{e|WT}$, (7), t_a is the aging duration .

$$t_a = T_s * \frac{\log P_{thresh}}{\log \gamma} \quad (8)$$

where γ is the decaying factor used in updating the delivery predictability, Ts is the beacon period, $n_{e|(t_a+T_c)}$ is number of nodes encountered during *ta + Tc* where *Tc* is the contact duration. Eqn (8) can be obtained easily from Eqn 1(c) by noting that the delivery predictability is updated every T_s seconds and that the value of P(a,b) is close to 1 when nodes a and b are in contact with each other for at least two beacons ($2T_s$ seconds) before they depart. Thus, $\log P_{thresh} = \log \gamma^k$, and $k = \frac{\log P_{thresh}}{\log \gamma}$. Since a beacon is transmitted every T_s seconds, and Eqn 1(b) is updated every beacon period, therefore Eqn(8) holds.

(b) determining p_e

Recall that p_e is the new delivery probability of a packet towards destination after being delivered to a node with the probability higher than P_{thresh}.

To calculate p_e, we rewrite the aging function as follows:

$$y = G(x) = \gamma^x \quad (9)$$

$$x = G^{-1}(y) \quad (10)$$

Eqn (9) assumes that the delivery predictability between two nodes immediately before disconnecting is close to 1 before it starts to age which holds most of the time when two nodes a,b are in contact for at least 2-3 beacon periods. Therefore,

$$p_e = E[G(x)] = \int_0^{G^{-1}(P_{thresh})} G(x) f(x)\,dx \quad (11)$$

where, *f(x)* is the probability density function of *x*. Since *x* is uniformly distributed between (0, $G^{-1}(P_{thresh})$) , we have f(x) = 1/$G^{-1}(P_{thresh})$. Hence,

$$p_e = \frac{1}{G^{-1}(P_{thresh})} \int_0^{G^{-1}(P_{thresh})} G(x)\,dx \quad (12)$$

(c) determining p_d

p_d is the new delivery probability of a packet towards destination after being delivered to a node at the expiration of *WT* (we assume in this case, the delivery probability is lower than P_{thresh}).

As before, we have,

$$p_{d'} = E[G(x)] = \int_{x=G^{-1}(P_{thresh})}^{\infty} G(x) f(x)\,dx \quad (13)$$

where, *f(x)* is probability density function of *x*, when $x \in (G^{-1}(P_{thresh}), EM)$ where *EM* is the Expected Meeting time derived in [2]. Therefore,

$$p_e = \frac{1}{EM - G^{-1}(P_{thresh})} \int_{G^{-1}(P_{thresh})}^{EM} G(x)\,dx \quad (14)$$

From [19],

$$EM = \frac{1}{1.75 p_m + 2(1-p_m)} \frac{N^2}{2R\overline{L}} (\overline{T} + \overline{T}_{stop}) \quad (15)$$

where $p_m = \frac{\overline{T}}{\overline{T} + \overline{T}_{stop}}$, $\overline{T}$ is epoch duration, $\overline{L}$ is epoch length, $\overline{T}_{stop}$ is the pause time.

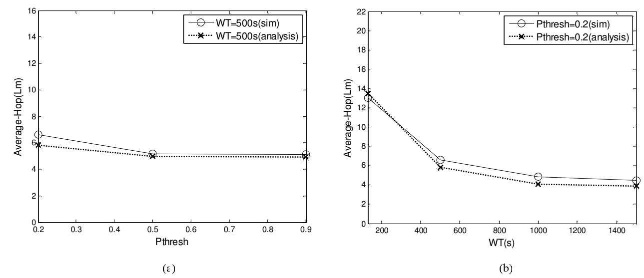


Figure 5: *Lm* obtained from analysis and simulations

We did some simulation studies to compare the average number of hops value seen in the simulation with the derived *Lm* value. We use a network scenario with 40 nodes distributed over 3000x3000 m^2. The nodes move according to the random waypoint model with speed randomly chosen between (1,5) m/s. The transmission range used is 250m. In the first set of experiment, we fix *WT* at 500s and vary P_{thresh} from 0.2 to 0.9. In Figure 5(a), we compare the analytical and simulation values for the expected hop count for this set of experiment. Our results show that the two values match closely. The results also show that when *WT* is set to 500s, setting P_{thresh} to 0.5 already allows us to reach the smallest expected hop count value. In our second set of experiment, we fix P_{thresh} to 0.2 and vary *WT*. The analytical and simulation values for the second experiment are plotted in Figure 5(b). Again, we see that the analytical and simulation values match closely.

B. *Approximate analysis for d_{hop}*

d_{hop} can be derived as follows:

$$d_{hop} = \sum_{i=1}^{n_e} q_i * Q(i) + \prod_{j=1}^{n_e} p_{0|j} * WT \quad (16)$$

where q_i is the probability of a packet is delivered to the i^{th} encounter during the *WT*,and *Q(i)* is the waiting time to meet this i^{th} encounter. Recall that $p_{0|j}$ can be computed using Eqn(7).

$$q_i = (1 - p_{0|i}) \prod_{j=1}^{i-1} p_{0|j} \quad \text{where i =1 ,…, } n_e \quad (17)$$

We approximate $Q(i)$ by assuming that the n_e encounter times are uniformly spread out within WT such that the average of the times of all encounters is 0.5WT. Thus, $Q(i)= i*WT/(1+ n_e)$. Our simulation results indicate that this is a relatively good approximation.

Thus,

$$d_{hop} = \sum_{i=1}^{n_e} q_i * i * WT/(1+n_e) + \prod_{j=1}^{n_e} p_{0|j} * WT \quad (18)$$

If we assume that the per hop delay is independent, then the average end to end delay, d_{e2e} ,will be Lm*d_{hop}. Since we assume sparse networks where the average internode encounter time is long, the independent per hop delay assumption is justified.

Next, we conduct some simulation studies to compare our analytical results with simulation results. We use the same network scenario used for Figure 5 (i.e. 40 nodes over 3000x3000m^2) . We use two multicast sessions, each with one source and 4 receivers. Each source generates 0.1 msg/sec. The source and the receivers of each session are randomly chosen among the 40 nodes. For each simulation run, we use a warmup period of 1000s before each flow starts generating traffic for 2000s, and the simulation continues to run until 10,000s. In Figures 6(a) and 6(b), we plot the average end-to-end delay we obtained from simulations versus the computed $Lm*d_{hop}$ value. The analytical results indicate that when WT increases, the average number of hops reduces but the average per-hop waiting time increases. The increasing per-hop waiting time with larger WT values means more buffers will be used. Thus, a tradeoff needs to be made. Given a P_{thresh} value, our analysis allows us to select a suitable WT value such that the EBMR scheme can achieve small Lm and per-hop delay values.

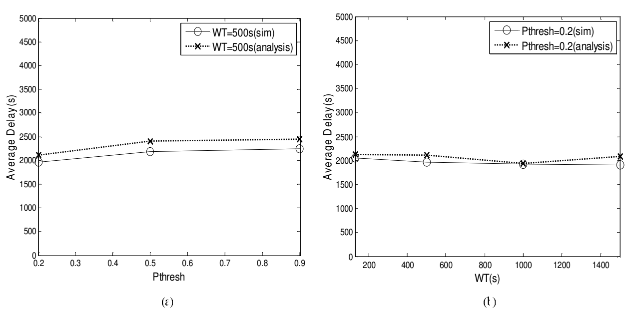


Figure 6: Avg E2E Delay with different P_{thresh} and WT values

V. SIMULATION STUDIES

A. Simulation Setup

In order to evaluate the EBMR scheme, we implement this scheme using NS-2 simulator [15]. In our simulation, the default network scenario is one where 40 nodes are randomly distributed over 3000x3000m^2. We assume that a 802.11 radio is used in each node. Thus, the default 802.11 parameters are used and the transmission range is set to 250m.

Mobility Model

By default, the nodes move according to the random waypoint model [16]. In the RWP model, each node randomly selects a destination location within the simulated geographical area, and move towards it using a constant speed v. Once it reaches the destination, it will pause for a certain time, then it repeats its action (i.e. picks another destination to move to). Unless otherwise stated, the pause time is set to 0 in our simulations and the node speed is chosen randomly between (1,5) m/s.

Traffic Model

For the traffic model, we use one multicast session with one source and multiple receivers. Each multicast source generates CBR traffic with a packet size of 512 bytes. We let the source generates traffic after 1000 seconds of warming up period and the traffic generation lasts for 2000 seconds but the simulation will run until 10,000 seconds. Each reported data point is the average of 10 runs.

Performance Metrics

The performance metrics we used in our evaluation are:

(1) *Delivery Ratio*, which is the ratio of the number of endpoints that receive a message and the number of intended receivers of the message [13]. This metric measures how successful the routing algorithm is in delivering the messages. For example, let NR be the number of receivers, s_i be the number of receivers that receive a message i, and T be the total number of multicast messages sent by the source, then $DR = \frac{\sum s_i}{T * NR}$,

(2) *Average Delay,* which is defined as the average end-to-end delay incurred by the delivered messages, and

(3) Data Efficiency, which is the total number of messages received divided by the number of transmissions used to deliver such messages [13].

B. *Comparison of different multicast routing schemes.*

In this section, we compare the following multicast delivery schemes: (a) brute-force unicast delivery using the original Prophet scheme where the source duplicates one message for each multicast receiver and each message is delivered using the original Prophet, (b) the early-decision based multicast strategy (denoted as Multicast), (c) EBMR delivery with $K=1$, $P_{threshold} = 0.5$ and $WT=500$ seconds (denoted as M-K=1), (d) EBMR delivery with $K=2$, $P_{threshold} = 0.5$ and $WT=500$ seconds (denoted as M-K=2). We use the default network scenario and let the nodes move according to the RWP model. Each multicast session has one source and 4 receivers. A multicast source generates 0.1 bundle every second. We vary the number of multicast sessions in this experiment.

The results for the delivery ratio, the average delay, the average number of hops it takes to reach a receiver, and the data efficiency are plotted in Figures 7(a),7(b), 7(c) and 7(d) respectively. Figure 7(a) shows that the M-K=2 scheme achieves the best delivery performance: it has the highest delivery ratio. The plot shows that the delivery ratio drops with increasing number of sessions with the Unicast approach. This can be explained as follows: for each session, the source node duplicates 4 copies per bundle so the total message rate is 0.4 bundle/s. As the number of sessions increases, the source nodes may also be used to relay traffic from other sessions. That increases further the total message rate that these nodes need to transmit. With an IFQ buffer size of 100, many packets are lost due to IFQ buffer overflows. Thus, the delivery ratio for the unicast approach drops. For other schemes, the total message rate that each source node needs to send out is in the range of 0.1-0.2 bundle/s and hence we see very few IFQ buffer overflows even with increasing number of sessions.

The lower average delay for the unicast and multicast approaches in Figure 7(b) is misleading since their delivery ratios drop significantly with increasing number of sessions, and only messages that take fewer hops can be delivered. The average delay for the M-K=1 scheme is higher than that for the Multicast scheme even though the M-K=1 scheme takes only 5 hops because the messages are queued longer at each intermediate node while waiting for better next-hop node when the M-K=1 scheme is used. The M-K=2 scheme achieves at least 25 to 30% lower average delay compared to the M-K=1 scheme because its average delay is the average delay of the first message copies that arrive at the receivers. Figure 7(c) shows that the average number of hops taken to reach the multicast receivers is significantly reduced with our enhanced EBMR scheme. Since K=2 incurs extra data redundancy, it is not surprising that the M-K=1 scheme achieves the best data efficiency as shown in Figure 7(d). The M-K=2 scheme still provides higher data efficiency than the Unicast and Multicast schemes.

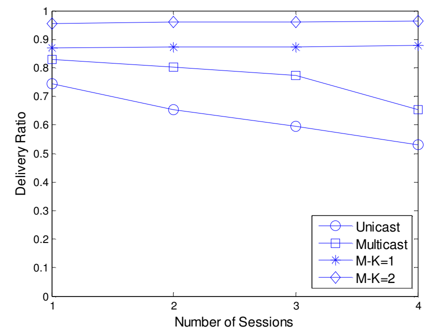


Figure 7(a) Delivery Ratio vs Number of Sessions

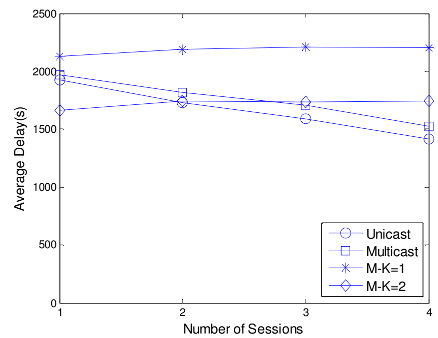


Figure 7(b): Average Delay vs Number of Sessions

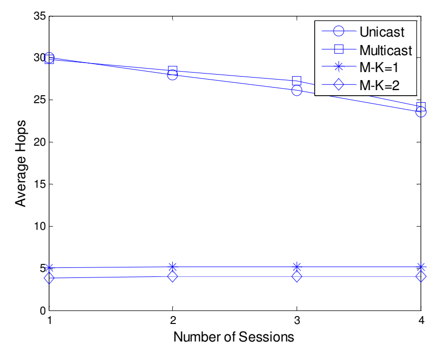


Figure 7(c): Avg Hops vs Avg Number of Sessions.

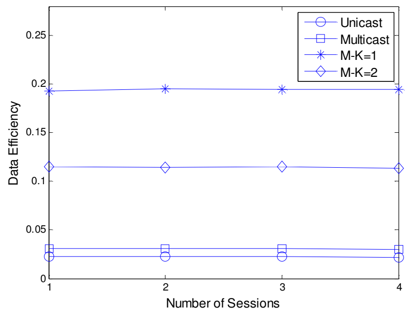


Figure 7(d): Average data efficiency vs Number of Sessions

VI. CONCLUDING REMARKS

In this paper, we have presented an encounter-based multicast routing scheme for DTNs. Our EBMR scheme allows nodes to cache the data until a good next-hop node can be found to relay the messages to the destinations. Via analysis, we have shown that with appropriate choice of P_{thresh} and WT, our EBMR scheme takes fewer number of hops to delivery multicast packets. Via simulation studies, we have demonstrated that this scheme can achieve high delivery ratio with reasonable data efficiency. An extended version of this paper which have additional simulation results can be found in [17].

There are several interesting issues we intend to explore further: we wish to investigate the impact of mobility models e.g. the Zebranet model [11], RPGM[6] etc, the impacts of having different multicast sessions, different number of multicast receivers, different node speeds etc on the delivery performance. In addition, we intend to explore an adaptive scheme where data replication is invoked only when the delivery predictability is below a certain threshold. We intend to implement this scheme on a medium size testbed to evaluate its performance with real multicast applications running in the DTN nodes.

ACKNOWLEDGMENT

This work has been supported by DARPA under Contract W15P7T-06-C-P430. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the sponsor of this work.

REFERENCES

[1] S. Das, C. Perkins, E. Royer, "Performance Comparision of Two On-Demand Routing Protocols", Proceedings of IEEE Infocom, March, 2000.

[2] T. Spyropoulos, "Performance Analysis of Mobility-assisted Routing", Proceedings of ACM MobiHoc, 2006.

[3] E. Royer, C. Perkins, "Multicast Adhoc On-Demand (MAODV) Routing", draft-ietf-manet-maodv-00.txt", July, 2000.

[4] J.J. Garcia-Luna-Aceves,"A multicast routing protocol for Adhoc Neworks (CAMP)", Proceedings of IEEE Infocom, 1999 pp 784-792.

[5] K. Fall, "A delay-tolerant network architecture for challenged internets", in SIGCOMM, 2003.

[6] X. Hong, M. Gerla, G. Pei, and C. C. Chiang, "A group mobility model for adhoc wireless networks", Proceedings of ACM International Workshop on Modeling, Analysis and Simulations of Wireless and Mobile Systems, (MSWiM), August, 1999.

[7] M.M.B.Tariq, M. Ammar, and E. Zegura, "Message Ferry Route Design for Sparse Ad hoc Networks with Mobile Nodes", ACM MobiHoc , May22-25, 2006.

[8] J. Burgess, B. Gallagher, D. Jensen, and B.L.Levine, Maxprop: Routing for vehicle-based disruption-tolerant networks. In INFOCOM, 2006.

[9] A. Lindgren, A.Doria, and O.Scheln, Probabilistic Routing in Intermittently Connected Networks. In Proc. Workshop on Service Assurance with Partial and Intermittent Resources, August 2004.

[10] Y. Wang, S. Jan, M. Martonosi, and K. Fall, "Erasure-Coding Based Routing for Opportunistic Networks", Proceedings of ACM Sigcomm WDTN Workshop, August 2005..

[11] A. Khelil et al, "Contact-Based Mobility Metrics for Delay-Tolerant Ad Hoc Networking", Proceedings of IEEE MASCOTS, 2005.

[12] W. Zhao, M. Ammar, E. Zegura, "Multicasting in delay tolerant networks: semnatics models and routing algorithms", Proceedings of ACM workshop on Wireless DTN, August , 2005.

[13] Q. Ye, L. Cheng, M. Chuah, B. Davison, "On-demand situation-ware multicasting in DTNs", Proceedings of IEEE Vehicular Technology Conference, May, 2006.

[14] P. Yang, M. Chuah, "Context-Aware Multicast Routing Schemes for DTNs", Proceedings of ACM Workshop on PE-WASUN, Aug, 2006.

[15] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research", Wireless Communications & Mobile Computing (WCMC), Vol .2, no. 5, pp. 483-502, 2002.

[16] "The network simulator ns-2", [Online] at http://www.isi.edu/nsnam/ns/

[17] Y. Xi, M. Chuah, "Performance Evaluation of an Encounter-Based Multicast Routing Scheme in DTNs", CSE Technical Report, Feb, 2008.