Multicasting in Delay Tolerant Networks: Semantic Models and Routing Algorithms

Wenrui Zhao, Mostafa Ammar and Ellen Zegura
College of Computing, Georgia Institute of Technology, Atlanta, Georgia 30332
{wrzhao, ammar, ewz}@cc.gatech.edu

***Abstract*— Delay tolerant networks (DTNs) are a class of emerging networks that experience frequent and long-duration partitions. These networks have a variety of applications in situations such as crisis environments and deep-space communication. In this paper, we study the problem of *multicasting* in DTNs. Multicast supports the distribution of data to a group of users, a service needed for many potential DTN applications. While multicasting in the Internet and mobile ad hoc networks has been studied extensively, due to the unique characteristic of frequent partitioning in DTNs, multicasting in DTNs is a considerably different and challenging problem. It not only requires new definitions of multicast semantics but also brings new issues to the design of routing algorithms. In this paper, we propose new semantic models for DTN multicast and develop several multicast routing algorithms with different routing strategies. We present a framework to evaluate these algorithms in DTNs. To the best of our knowledge, this is the first study of multicasting in DTNs. Our objectives are to understand how routing performance is affected by the availability of knowledge about network topology and group membership and to guide the design of DTN routing protocols. Using *ns* simulations, we find that efficient multicast routing for DTNs can be constructed using only partial knowledge. In addition, accurate topology information is generally more important in routing than up-to-date membership information. We also find that routing algorithms that forward data along multiple paths achieve better delivery ratios, especially when available knowledge is limited.**

I. INTRODUCTION

Delay tolerant networks (DTNs) are a class of emerging networks that experience frequent and long-duration partitions [17], [21]. There is no end-to-end path between some or all nodes in a DTN. These networks have a variety of applications in situations that include crisis environments like emergency response and military battlefields, deep-space communication, vehicular communication, and non-interactive Internet access in rural areas [1], [3], [9], [10], [19], [24], [29], [32], [33], [35], [37].

In this paper, we study the problem of *multicasting* in delay tolerant networks. Multicast service supports the distribution of data to a group of users. Many potential DTN applications operate in a group-based manner and require efficient network support for group communication. For example, in a disaster recovery scene, it is vital to disseminate information about victims and potential hazards among rescue workers. In a battlefield, soldiers in a squad need to inform each other about their surrounding environment. Although group communication can be implemented by sending a separate copy of data via unicast to each user, this approach suffers from poor performance, which is confirmed in our simulations. The situation is especially acute in DTNs where resources such as connectivity among nodes, available bandwidth and storage are generally severely limited. Thus efficient multicast services are necessary for supporting these applications.

Multicasting in the Internet and mobile ad hoc networks (MANETs) has been studied extensively in the past [5], [15], [16], [26], [12], [25], [28] (see [4], [14] for surveys on these topics). However, due to the unique characteristic of frequent partitioning in DTNs, multicasting in DTNs is a considerably different and challenging problem. It not only requires new definitions of multicast semantics but also brings new issues to the design of routing algorithms.

The semantics of multicasting in traditional networks such as the Internet and MANETs are straightforward, specifying that packets sent to a multicast group be delivered to members of the group. Since data transfer delay in these networks is short (on the order of milliseconds), group membership changes during data transfer are rare and can be ignored. Thus the receivers of a multicast packet are well defined, i.e., all current group members. This, however, is no longer valid in DTNs. Due to frequent partitions and consequently large transfer delays in a DTN, membership changes during data transfer are the norm rather than the exception. Under these situations, it is not obvious how to define the receivers of a multicast packet, relative to the group membership over time.

Consider a simple example where a source sends a message to a group at time t. Let t' be the earliest time that other nodes could possibly receive this message according to network topology limitations. Suppose that node A joins the group at time $t_1 < t$ and leaves at time t_2, $t < t_2 < t'$. Node B joins at time t_3, $t < t_3 < t'$ and never leaves. From the perspective of traditional multicasting, it is not clear which nodes should receive this message, whether A, B, both or neither of them. For node A, it is a group member at the time of message generation but no longer a member at the earliest time of potential message delivery. The reverse is true for node B. To address this problem, new semantic models are needed for DTN multicasting.

In this paper, we develop new multicast semantic models for DTN environments that have explicit constraints on group membership and delivery action. These semantic models unambiguously define the receivers of a multicast packet and have various applications in DTN environments.

With these semantic models, we study the problem of multicast routing in DTNs. DTNs introduce several challenges for routing. First, there may be no end-to-end path between nodes in DTNs. Traditional routing algorithms would fail to deliver

This work was supported by the NSF Grant ITR-0313062 and AFOSR MURI Grant F49620-00-1-0327.

data because no route is found to reach the destinations. Thus multicast routing in DTNs needs to operate in the presence of network partitions. Second, as proposed in [17], data transfer in DTNs is in application data units called *messages* (or bundles). This is different from the use of flows in traditional multicasting. Third, information about nodes joining or leaving a group may be available to nodes only after significant delays because of network partitions. Multicast routing algorithms need to handle these highly delayed join or leave requests. Finally, the multicast semantic models developed in this paper also introduce new requirements for message forwarding.

We study four classes of multicast routing algorithms for DTNs with different routing strategies. To understand routing performance in DTN environments where available routing information may be significantly limited by network partitions, we present an evaluation framework that models different levels of available knowledge about network topology and group membership. This is an extension of the framework for unicast routing developed in [21]. Our objectives are to understand the impact of the availability of knowledge on routing performance and to guide the design of DTN routing protocols. With extensive *ns* [27] simulations, we evaluate various routing algorithms. We find that efficient routing for multicast can be constructed using only partial knowledge, as in the case of unicast [21]. In addition, accurate topology information is generally more important in routing than up-to-date membership information. Furthermore, routing algorithms that forward data along multiple paths achieve better delivery ratios, especially when available knowledge is limited. Finally, our results confirm that unicast-based approaches that send a separate copy of messages to each receiver perform poorly in DTNs.

The rest of this paper is structured as follows. Section II describes the network model and briefly reviews unicast routing in DTNs. In Section III, we present new semantic models for DTN multicasting. We describe an evaluation framework for multicast routing in Section IV. Section V presents the four classes of multicast routing algorithms. We present simulation results in Section VI and review related work in Section VII. The paper is concluded in Section VIII.

II. DTN Network Model

In this section, we present the network model considered in this paper and briefly review unicast routing in DTNs.

A. Network Model

We assume that nodes in DTNs are identified by a unique ID. An *endpoint* is an entity at a node that acts as the source or destination of communication, e.g., an application at the node. An endpoint is identified by an *endpoint ID* which is a tuple $(node\_id, entity\_id)$ where $entity\_id$ uniquely identifies an endpoint within a node[1]. We assume a message-oriented service, where endpoints communicate using application data units called *messages*. In addition, nodes are assumed to have synchronized clocks, which could be loose depending on application requirements.

In our model, multicasting disseminates messages to a group of endpoints that are identified by a *group ID*. A group ID is a globally unique ID that has the same form as endpoint IDs. A multicast message encodes a group ID as the destination endpoint, which can be distinguished from unicast endpoint IDs. In order to receive messages destined to a specific group, endpoints join the group by indicating a JOIN request with the group ID to the DTN routing agent at the node. Similarly, an endpoint leaves a group using a LEAVE request to stop receiving messages for a group. Routing agents in a DTN may authenticate endpoints and authorize JOIN or LEAVE requests according to administrative policies. However, in this paper, we consider a general multicast model in which endpoints can join and leave groups autonomously.

In a DTN, network partitions may occur frequently. To overcome disconnections, data is forwarded in a *store-carry-and-forward* fashion, i.e., a node buffers messages in its storage until connections with other nodes become available. We assume that node storage is used for holding in-transit messages only. Delivered messages are stored in separate application buffers.

B. Unicast Routing in DTNs

We now briefly describe the unicast routing in DTNs developed by Jain et al. [21]. A DTN is represented as a directed multi-graph. Thus there may exist multiple edges between two nodes. Each edge represents a connection between nodes and has time-varying capacity and propagation delay that represent the properties of the connection over time. The capacity of an edge is zero when the corresponding connection is unavailable. A *contact* is defined as an opportunity to send data between nodes, i.e., an edge and the time interval during which the edge capacity is positive. Fig. 1 shows a DTN graph in which there are three edges between node A and node B. Contacts are shown along each edge with their time intervals. We can see that there are two contacts for edge e_1, from time 5 to 10 and from time 50 to 60 respectively.

Given the time-varying capacity and delay of edges in a DTN, routing decisions vary with time. Suppose that node A sends messages to node B and the routing objective is to minimize the message transfer delay. For simplicity of presentation, we ignore the transmission delay and propagation delay at the edges. If a message arrives at node A at time 0, the optimal route to node B is via $contact1$ of edge e_1 which has the minimum delay of 5. If a message arrives at time 15, however, the optimal route would be via $contact3$ of edge e_2 since $contact1$ is no longer available. To compute the shortest (or minimum delay) paths in a DTN graph, Jain et al. [21] develop a modified Dijkstra's algorithm. The key difference in the modified algorithm is to take into account of the time of message arrivals at each node and only consider contact opportunities after message arrivals. In this paper, we assume that this algorithm is used to compute routes in a DTN graph and use the term "shortest path" and "minimum delay" of a message to refer to the path computed by this algorithm and

[1]Other addressing schemes are certainly possible, however, we focus on this fairly standard scheme in this paper.

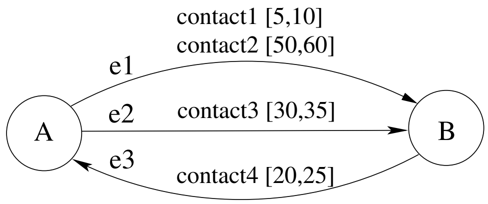


Fig. 1. An example of DTN graphs.

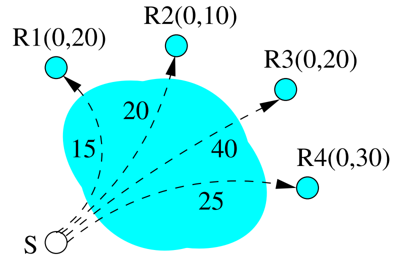


Fig. 2. An example of multicast semantic models. The figure shows a DTN at time 0 when a message for group G is generated at node S. The dashed lines are the shortest paths from S to other endpoints with the minimum delay shown along each path. The time interval during which an endpoint is a member of group G is shown next to the endpoint.

the corresponding delay of forwarding the message along the computed path respectively. Readers should refer to [21] for more details about this algorithm.

III. Multicast Semantic Models

In this section, we will present three new semantic models for multicasting in DTNs[2]. As discussed earlier, due to large transfer delays in DTNs, group membership may change during a message transfer, introducing ambiguity in multicast semantics. Under these situations, it is necessary to make a distinction between *group members* and the *intended receivers* of a message, i.e., endpoints to which the message should be delivered. Group members may change with time as endpoints join and leave the group. The intended receivers, on the other hand, should be fixed for a message, even though they are defined based on group membership. We develop three multicast semantic models that allow users to explicitly specify temporal constraints on group membership, unambiguously defining the intended receivers of a message. These models also specify constraints on the action of message delivery and have important applications in DTN environments.

A. Temporal Membership Model

To determine the receivers of a multicast message, we need to explicitly specify the time during which the intended receivers are defined. One straightforward approach is to define the receivers of a message as the group members at the time of message generation. In this paper, we consider a more general semantic model, called *Temporal Membership (TM)*, which gives users explicit control over the time-based definition of group membership. In the TM model, a message includes a *membership interval* that specifies the period during which the group members are defined. For a message with group ID G and membership interval $[t_1,t_2]$, the intended receivers of the message consist of endpoints who are members of group G at *any* time during period $[t_1,t_2]$[3]. Under the TM model, the receivers of a message are well defined. By setting the membership interval to $[t_0,t_0]$ where t_0 is the message generation time, the receivers are the group members at the time of message generation. In the TM model, there is no delivery constraint so messages can be delivered at any time. Note that the intended receivers of a message may be different from the *actual receivers* which actually receive the message. The actual receivers are a subset of the intended receivers and dependent on the routing algorithm used and the traffic condition in the network.

The TM model allows users to flexibly specify the time-based characteristics of the receiving group of a message, which has some interesting applications in DTNs. One potential application of the TM model is in mobile sensor networks where mobile sensors record sensory data along their movement trajectories. Each region of interest is associated with a multicast group and sensor nodes join or leave multicast groups based on their locations. To query the status of a given region during a specific period, a user can send a multicast message to the group that is associated with the region with a specific membership interval. The message will be delivered to sensors that are in the region during the specified period. To implement the TM model, a message can encode the type of semantic model and the membership interval in the message header.

Consider an example in Fig. 2 which shows a DTN at time 0 when a message is generated at node S. If the membership interval of the message is $[0,1]$, the intended receivers are $\{R_1,R_2,R_3,R_4\}$. If the membership interval is $[15,20]$, the intended receivers become $\{R_1,R_3,R_4\}$ since R_2 is no longer a group member during this period.

B. Temporal Delivery Model

Our second model is the *Temporal Delivery (TD)* model. In this model, messages specify additional constraints on the action of message delivery beyond the unconstrained TM model. A message specifies both a membership interval and a *delivery interval*. The delivery interval indicates the time period during which the message should be delivered to the intended receivers, as will be defined below. Note that the message can be delivered to nodes hosting the intended receivers *before* that period since nodes can delay forwarding the message to endpoints[4].

[2]Other models are possible, however these models seem to capture the needs of many applications. Further experience with DTN applications will help clarify which semantics are most useful.

[3]An alternative and complementary model would require endpoints to be group members throughout period $[t_1,t_2]$. As an initial effort, we focus on the TM model in this paper.

[4]An alternative model would require the message be delivered to nodes hosting the intended receivers only during the delivery interval. Under this model, a node can act as a relay for messages only when there is no receiver at the node. This requires that either nodes have knowledge about future group membership of local endpoints, or all relaying nodes are not data sources or destinations.

To be consistent with this delivery constraint, the intended receivers of a message should *exclude* endpoints that are not able to receive the message during the delivery interval. Let R be the set of all endpoints in the network and $member(r,t',t'')$ be a predicate on whether endpoint r is a group member during period $[t',t'']$. Let t_0 be the message generation time and $d(t,r)$ be the minimum delay from the source of the message to endpoint r starting at time t. As described in Section II, $d(t,r)$ can be computed using the modified Dijkstra's algorithm in [21]. For a message with group ID G, membership interval $[t_1,t_2]$ and delivery interval $[t_3,t_4]$, the set $I_{\mathcal{TD}}$ of intended receivers is defined as

$$I_{\mathcal{TD}} = \{r | member(r,t_1,t_2) = true \textbf{ and } d(t_0,r)+t_0 < t_4, r \in R\}. \quad (1)$$

Note that while the delivery interval specifies that the message be delivered no earlier than t_3, the definition of $I_{\mathcal{TD}}$ does not require an earliest time for the message to reach a node. This is because nodes can delay forwarding the message to endpoints. The TD model is more general than the TM model, which is a special case with delivery interval $[t_0,\infty)$.

The TD model enables users to have additional control on when messages are delivered. In addition, a delivery interval specifies an expiration time for a message. This enables routing algorithms to remove messages that are not able to meet the delivery intervals and reclaim storage space, which is crucial in DTNs since nodes may need to buffer messages for a significantly long period.

Consider an example using Fig. 2. For a message with membership interval $[0,1]$ and delivery interval $[0,35]$, the intended receivers are $\{R_1,R_2,R_4\}$. R_3 does not meet the delivery interval since it could receive the message no earlier than time 40, hence R_3 is not an intended receiver of this message.

C. Current-Member Delivery Model

In both the TM and TD models, receivers of a message are not required to be group members at the time of message delivery. In our third model, the *Current-Member Delivery (CMD)* model, messages explicitly specify whether this requirement should be met. A message includes a *CMD flag* as well as a membership interval and a delivery interval. When the CMD flag is set, the receivers of the message should be group members at the time of message delivery. In addition, the message should be delivered during the delivery interval as in the TD model. When the CMD flag is not set, the CMD model reduces to the TD model, thus the CMD model is a more general model. Fig. 3 depicts the relationship among these semantic models.

We now define the intended receivers $I_{\mathcal{CMD}}$ of a message in the CMD model. When the CMD flag is set, $I_{\mathcal{CMD}}$ should exclude endpoints that are not able to be group members at the time of message delivery. Using the same notations as in the previous section, we define $I_{\mathcal{CMD}}$ as follows

$$I_{\mathcal{CMD}} = \{r | r \in I_{\mathcal{TD}} \textbf{ and } member(r,t_m,t_4) = true\} \quad (2)$$

where t_m is $\max(d(t_0,r)+t_0,t_3)$, the earliest time that the message could be delivered to endpoint r because of the transfer delay from the source to r and the delivery interval constraint. In order to meet the CMD constraint, r must be a group member during period $[t_m,t_4]$.

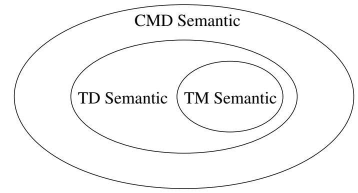


Fig. 3. DTN multicast semantic models.

Consider an example using Fig. 2. For a message with membership interval $[0,1]$, delivery interval $[0,35]$ and the CMD flag set, the intended receivers are $\{R_1,R_4\}$. R_2 is not an intended receiver because it could not be a group member at the time of message delivery which is at least time 20.

IV. Multicast Routing Framework

Given these semantic models, we now turn to the problem of multicast routing in DTNs. In this section, we first discuss the routing objectives in DTN multicasting. We then present a framework for evaluating multicast routing in DTNs. This is an extension of the framework for unicast routing developed in [21]. In our study, routing algorithms use various information about network conditions to achieve better performance. Due to network partitions in DTNs, however, there might not be complete or current knowledge available, degrading routing performance. In this paper, we study the fundamental trade-off between the amount of available knowledge and the achieved performance. To model the availability of knowledge, we use abstract *knowledge oracles* that encapsulate particular knowledge about network status to be used in routing algorithms [21]. While the use of knowledge oracles does not consider how such knowledge is actually disseminated in the network and how much overhead it causes, this approach isolates the effects of knowledge availability on routing performance, which would provide insight to guide the design of routing protocols. We are currently working on the problem of control information dissemination in DTNs.

We consider two types of knowledge that might have direct effect on routing performance, knowledge about contact opportunities and knowledge about group membership. We describe various contact and membership oracles, which represent different situations ranging from no knowledge, partial knowledge, to complete knowledge. We finally present an overview of four approaches of multicast routing for DTNs.

A. Routing Objectives

For any routing algorithm, a basic objective is to maximize the probability of delivering messages. For multicasting, a message is sent to a group of receivers. A message might not be able to reach all receivers because of message drops at times of storage shortage and routing loops. This is more evident in DTNs which are characterized by frequent partitions. In this paper, we evaluate multicast routing algorithms by the *message delivery ratio* which is the ratio between the number of endpoints that receive a message and the number of intended receivers of the message according to the semantic model used. This metric measures how successful the routing algorithm is in delivering messages.

The delivery ratio, however, does not reflect the efficiency of a routing algorithm in utilizing resources. A routing algorithm might achieve higher delivery ratios at the cost of generating much more traffic in the network. We define the *routing efficiency* of an algorithm as the ratio between the total amount of delivered messages and the total amount of traffic generated in the network. This metric is especially important for mobile networks where power supplies are limited. Depending on application environments, routing algorithms need to achieve a different balance between delivery ratio and routing efficiency. Another metric we consider is the *average message delay*, i.e., the average time from the generation of a message to the reception of the message at a receiver. Although emerging DTN applications are expected to be able to tolerate delay in message transfer, many applications will benefit from reduced delay.

In this paper, we study several classes of routing algorithms that are expected to achieve different balance of delivery ratio, routing efficiency and robustness again incomplete knowledge. For each class of algorithms, we focus on minimizing the delay for each intended receiver.

B. Contact Knowledge Oracles

We consider three levels of knowledge about network topology using the following oracles, which are developed in [21].

- *Null Contact Oracle.* This oracle can not answer any question about contact opportunities. That is, this oracle represents no knowledge about network topology.
- *Contact Summary Oracle.* This oracle can answer questions about the long-term statistics regarding network topology. Specifically, it can provide the average time between contact occurrences and average contact duration. This oracle represents partial knowledge about network topology.
- *Complete Contact Oracle.* This oracle can answer any question about network topology, even topology at a future time. Specifically, it can provide the exact time when a contact occurs, the duration, capacity and delay of the contact. This represents the availability of complete knowledge about network topology.

C. Group Membership Knowledge Oracles

Group membership oracles answer questions about group dynamics, e.g., the events of an endpoint joining or leaving a group. We consider the following group membership oracles.

- *Local Membership Oracle.* This oracle can answer questions of a node about group membership of endpoints at the node up to the current time but provide no information on membership for endpoints at other nodes.
- *Delayed Membership Oracle.* In DTNs, the dissemination of information is delayed by network partitions. To model the availability of membership knowledge when membership information is communicated in-band, we define the delayed membership oracle as follows. For an endpoint r and a node S that queries the oracle, this oracle can answer questions about membership of endpoint r up to a specific time t. t is the latest time that satisfies $d(t,S)+t \le t_0$ where t_0 is the current time and $d(t,S)$ is the minimum delay from endpoint r to node S starting at time t[5]. In other words, if endpoint r joins or leaves a group at or before time t and sends this information to other nodes by flooding, assuming no contending traffic in the network, node S should have received this information by the time of querying the oracle.
- *Current Membership Oracle.* This oracle can answer questions about group membership of all nodes up to the current time.
- *Complete Membership Oracle.* This oracle can answer questions about group membership of all nodes at any time. It represents an ideal and largely unrealistic scenario where complete knowledge about group membership is available for all time.

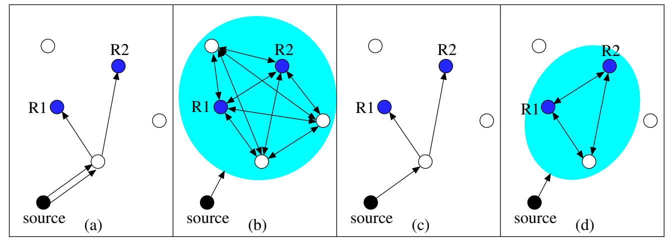


Fig. 4. Routing approaches in DTNs. (a) unicast-based routing (b) broadcast-based routing (c) tree-based routing (d) group-based routing.

D. Routing Approaches

We now describe four approaches for multicast routing in DTNs which are common for multicasting in the Internet or MANETs. These approachs have different properties in delivery ratio, routing efficiency and robustness against incomplete knowledge. In this paper, we aim to study how various approaches perform in DTNs. In the next section, we will describe how to design routing algorithms in DTNs that are based on these routing approaches and various knowledge oracles.

1. Unicast-based routing (UBR)

This approach implements multicast service by using unicast routing in DTNs. Specifically, the source will send a copy of the message to every intended receiver. Because of duplicate transmissions, this approach typically has lower routing efficiency and delivery ratios, especially when the number of receivers in the group is large. Fig. 4(a) shows an example of UBR.

2. Broadcast-based routing (BBR)

In broadcast-based routing or epidemic routing [32], messages will be flooded throughout the network in order to

[5]The minimum delay depends on the message size since it affects the transmission delay. In the delayed membership oracle, we try to model the availability of knowledge that is limited by the network topology. Thus we are not concerned about the actual transmission of membership information and use a message size of zero in computing this delay.

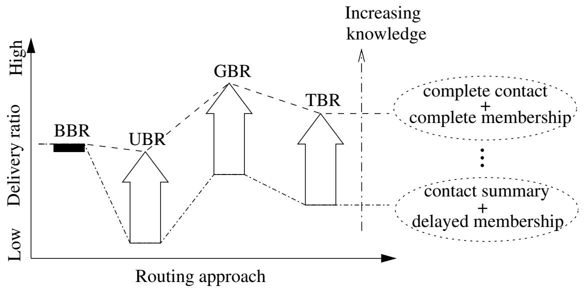


Fig. 5. Conceptual performance of various routing approaches under different levels of knowledge.

reach the intended receivers. This approach does not require knowledge about contacts or group membership, thus is very robust against the lack or inaccuracy of knowledge. On the other hand, this approach has low routing efficiency since messages are often unnecessarily forwarded to many nodes. Fig. 4(b) shows an example of BBR.

3. Tree-based routing (TBR)

In TBR, messages are forwarded along a tree in the DTN graph that is rooted at the source and reaches all receivers. Messages are duplicated only at branching nodes that have more than one outgoing path. While this looks similar to traditional multicast protocols that use multicast trees, routing in TBR occurs in a "store-carry-and-forward" fashion that may buffer messages for a significant period of time. Fig. 4(c) shows an example of this approach. TBR is expected to have high routing efficiency. On the other hand, forwarding messages along a single path to a receiver is not robust against inaccurate or partial knowledge, or traffic variation in the network, which often leads to low delivery ratios.

4. Group-based routing (GBR)

GBR uses the concept of *forwarding group* [12], [25] to address the problems of BBR and TBR. A forwarding group for a message is a set of nodes that forward the message via flooding among themselves. More specifically, GBR determines a forwarding tree to reach all receivers and sets the forwarding group as nodes in the forwarding tree, including the receivers. Messages in GBR will be flooded within the forwarding group to increase the chance of delivery. Thus GBR can be seen as a hybrid between BBR and TBR. Fig. 4(d) shows an example of GBR. As compared to TBR, GBR is expected to have higher delivery ratio and lower efficiency since GBR floods messages among nodes in the tree. As compared to BBR, GBR is less robust to partial knowledge but has higher efficiency.

Fig. 5 summarizes the conceptual performance of various routing approaches under different levels of available knowledge. BBR is expected to achieve the same delivery ratio under different amount of available knowledge. The expected delivery ratio of other approaches (i.e., UBR, GBR and TBR) would improve with the increasing knowledge. These approaches would also achieve different routing efficiency.

V. Multicast Routing Algorithms

In this section, we will develop multicast routing algorithms for DTNs based on the four routing approaches and various knowledge oracles described in the previous section. Due to the unique characteristics of DTNs, these algorithms differ from those of wired networks or MANETs in the following aspects. First, data is forwarded in the unit of messages. Consequently, nodes maintain forwarding states on a per message basis. Second, to overcome network partitions and handle delayed join requests, messages will be buffered in node storage for an extended period, even after being forwarded to all next hops. Third, nodes exchange control information to determine the set of messages that should be forwarded. This negotiation procedure avoids duplicate message transmissions and prevents routing loops that might arise due to the dynamics of network topology or the use of message flooding.

The general operations of these algorithms are as follows. When a message arrives, either generated by an endpoint or received from another node, a node first determines if there is any local intended receiver of the message. If so, the message is forwarded to these receivers according to the semantic model, as will be described in Section V-B. The message is then buffered in node storage and forwarded to other nodes when contacts become available. To determine the routes or the next hops for a message, nodes maintain local forwarding state for each buffered message, which is computed based on the available knowledge of contacts and group membership. Nodes update the forwarding state as group membership changes, e.g., endpoints joining or leaving a group.

In DTN multicast routing, messages will be buffered in node storage until being deleted due to buffer overflows or being expired according to the semantic models. Thus nodes other than the source can handle join requests and send buffered messages to new receivers. As compared to approaches where only the source handles join/leave requests, this approach has the advantages of reducing message delay and improving message availability. In this paper we adopt an age-based buffering policy which removes the oldest message when the buffer overflows, thus giving new messages opportunities to be delivered.

In the following, we will first describe the common operations of these algorithms and then the specifics for each algorithm.

A. Message Forwarding between Nodes

In this section, we describe how multicast messages are forwarded between nodes in a DTN, as is illustrated in Fig. 6. We first explain how forwarding state is maintained for each message. As described above, a newly arrived message will be forwarded to receivers at the node and inserted into node storage. Every node maintains local forwarding state for each message buffered in node storage. Each message has a NEXT-HOP list $\mathcal{L}_n$ that records nodes to which this message should be sent. $\mathcal{L}_n$ is initialized upon the message arrival and updated when group membership changes. The determination of $\mathcal{L}_n$ is different in various algorithms and will be described in Section V-C to V-G. Each message is also associated with a SENT list $\mathcal{L}_s$ that consists of nodes that have already received this message. $\mathcal{L}_s$ is empty for new messages. In the rest of this paper, we use $\mathcal{L}_n(m)$ and $\mathcal{L}_s(m)$ to denote the NEXT-HOP and SENT lists for message m respectively.

1. On arrival of message m
   Forward m to local receivers if any;
   Insert m in node storage;
   Initialize forwarding state $\mathcal{L}_n(m)$ and $\mathcal{L}_s(m)$;
2. On contact with node B
   For each message m in storage
      IF $B \in \mathcal{L}_n(m)$ and $B \notin \mathcal{L}_s(m)$
         Send a copy of m to node B;
         Add B into $\mathcal{L}_s(m)$;
3. On join/leave request for group G
   Update $\mathcal{L}_n$ for messages destined for G;

Fig. 6. Message forwarding between nodes.

To compute $\mathcal{L}_n$, nodes need to estimate the intended receivers of a message based on the current available knowledge. As more knowledge about group membership becomes available, the estimated intended receivers may change. Specifically, when being informed of endpoint r joining a group, a node needs to determine whether r is an intended receiver for messages buffered in node storage. For the TM model, r would be an intended receiver if the join time is within the specified membership interval. For the TD and CMD models, r needs to meet additional delivery constraints as defined in (1) and (2) respectively. Both (1) and (2) require computing the *member* predicate and the minimum delay $d(t,r)$. However, nodes may not be able to compute the *member* predicate since the required membership information is currently not available to the node, e.g., future membership information when the delayed membership oracle is used. Similarly, the minimum delay $d(t,r)$ may not be determined if there is no exact contact information available, e.g., using the contact summary oracle. In these situations, nodes will ignore those requirements that can not be determined. For the case of group leave events, a node can estimate the intended receivers similarly. With the estimated intended receivers, nodes can computed $\mathcal{L}_n$ for each message according to the routing approach used, such as BBR, UBR, TBR and GBR.

Given the message forwarding state, nodes forward messages as follows. Suppose that a contact between node A and B becomes available. For each buffered message m, node A will try to forward message m to node B if B is in $\mathcal{L}_n(m)$ and not in $\mathcal{L}_s(m)$. In other words, node B should be a next hop for this message and node A has not transmitted this message to node B before. After transmission, node A will add node B into $\mathcal{L}_s(m)$. So node A will not send duplicate messages to node B.

Despite the use of $\mathcal{L}_s$, nodes may still receive duplicate messages from different nodes. This may happen when there is a routing loop in the network due to the dynamics of network topology. Or the algorithm (e.g., GBR or BBR) uses flooding to forward messages. Duplicate transmissions waste valuable bandwidth in DTNs. To address this problem, nodes may exchange control information first to determine which messages should be sent. Here we assume that the delay of the contact is small as compared to the duration of the contact[6]. Suppose that node A tries to transmit messages to a neighbor B. Node A will first send an ADV message including information about messages it wants to transmit. Upon reception of the ADV message, node B replies with a REQ message which lists only messages it currently does not have. Then node A will send the messages listed in the REQ message. Since the size of meta data can be much smaller than that of the actual message, the overhead of ADV and REQ messages is generally not significant.

[6]If contacts are short in duration, other strategies are needed.

B. Local Message Forwarding

We now turn to the problem of how messages are forwarded to endpoints at each node according to the delivery constraints of the semantic model, which may require nodes to delay message forwarding. Specifically, for a newly arrived message, a node identifies endpoints at the node that meet the specified membership interval. For each of these endpoints, say r, the node performs local forwarding as follows. If the TM model is specified, the node will pass the message to r immediately since there is no delivery constraint. When the TD model is used, the message will be forwarded to r if the current time meets the delivery interval. Otherwise, the node will delay delivering the message to r or drop the message depending on whether the current time is earlier or later than the delivery interval. When the CMD model is specified and the CMD flag is set for the message, no action is needed if r is currently not a member; otherwise, it follows the same procedure as the case when the TD model is used.

Group joins and leaves may affect this local forwarding procedure, but only for messages using the CMD model and with the CMD flag set. In this case, when a local endpoint that meets the membership constraint of a message joins the group, it becomes a group member. So the message will be forwarded according to the delivery interval specified. Similarly, when a local endpoint that meets the membership constraint of a message leaves the group, the message will not be forwarded to this endpoint.

In the following sections, we will present the various multicast routing algorithms, starting with two variants of the Tree-Based Routing approach.

C. Static Tree-Based Routing

In TBR, nodes forward messages along a forwarding tree in the DTN graph. We first describe the Static Tree-Based Routing (STBR) which uses a forwarding tree that is static for a given set of receivers. So the paths to receivers are independent of other traffic in the network. Specifically, nodes construct a shortest path tree from the source to the estimated intended receivers of a message starting at the message generation time. Note that the forwarding tree may change as the estimated intended receivers change with endpoints joining or leaving the group. Messages are then forwarded along the tree, i.e., the NEXT-HOP list $\mathcal{L}_n$ includes nodes that are the next hops in the tree. $\mathcal{L}_n$ is computed at the time of message arrivals and when group membership changes.

We now describe how a node computes $\mathcal{L}_n$ for a message. As a message m arrives, $\mathcal{L}_n(m)$ is initially set to empty. STBR then uses the modified Dijkstra's algorithm in [21] to compute the shortest path from the message source to each estimated intended receiver starting at the message generation time. If the node itself is in the shortest path, the next hop in this path is inserted into $\mathcal{L}_n(m)$. Note that the modified Dijkstra's algorithm can be used when either the complete contact or contact summary oracle is used. In the case when the complete contact oracle is used, the DTN graph reflects the exact contacts between nodes. In the case when the contact summary oracle is used, only long-term statistics regarding the network topology are available. The delay of an edge in the DTN graph thus is set to the sum of the average queuing delay, propagation delay and transmission delay of the corresponding contact. As a result, the DTN graph and shortest paths are time-invariant when the contact summary oracle is used.

In STBR, the route from the source to an intended receiver is static. Thus if a message misses a contact with a node in $\mathcal{L}_n$, the message needs to wait for the next opportunity to connect to this node, which may significantly increase the message delay. In addition, the use of static routes disallows nodes to utilize local or more accurate information to forward messages along better paths, i.e., to avoid congestion in a path or to use other available contact opportunities. This becomes more evident when only limited knowledge is available.

D. Dynamic Tree-Based Routing

To address the above problems with static TBR, we develop the Dynamic Tree-Based Routing (DTBR) algorithm. DTBR uses explicit addressing [2], i.e., messages include the endpoint IDs of the receivers as well as the group ID in the message header. In the following, we use the term "explicit address(es)" to refer to these explicitly addressed endpoint IDs. With explicit addressing, nodes can determine the next-hops of a message dynamically based on current available information, such as local queuing information or newly available contact information. Specifically, DTBR determines the NEXT-HOP list $\mathcal{L}_n$ by computing the shortest paths from the current node to endpoints embedded in the message. Nodes which are the next hops in these paths are added to $\mathcal{L}_n$. This is in contrast with STBR which uses the shortest paths from the source at the message generation time to all intended receivers.

We now describe how DTBR maintains the explicit addresses in a message. The explicit addresses of a message are initially set to the estimated intended receivers. When a message is duplicated and forwarded to another node or to an endpoint at the node, the explicit addresses are split such that each copy of the message contains only the IDs of endpoints to which it will be delivered. In addition, DTBR needs to add new endpoints to the explicit addresses of a message when endpoints join the group and become new intended receivers. Since multiple nodes may buffer the message and be able to send this message to the new receiver, this raises the problem of which nodes should take this responsibility. In this paper, we adopt the following approach to address this problem. Suppose that the joining endpoint is r and becomes a new intended receiver of a message. When informed of this join event, each node computes the shortest path from the source of the message to r starting at the message generation time. If the node is on this shortest path, it will be responsible for sending the message to endpoint r, i.e., inserting r to the explicit addresses of the message. While this might still result in multiple nodes sending the message to endpoint r, the number of such nodes is no more than the number of nodes in the computed shortest path. In addition, the use of ADV/REQ messages would reduce duplicate message transmissions.

Explicit addressing increases the size of messages. However, the effect on performance would be modest when the size of the original message is large or the number of receivers is small.

E. Group-Based Routing

In Group-Based Routing (GBR), nodes construct a forwarding group for each message which forwards the message to the estimated intended receivers. Specifically, GBR computes the shortest path tree as in STBR and sets the forwarding group as the set of nodes in the tree including the receivers. Messages are then forwarded by flooding within the forwarding group. Thus GBR is the same as STBR except in the determination of $\mathcal{L}_n$. In GBR, $\mathcal{L}_n$ consists of all nodes in the shortest path tree while in STBR, $\mathcal{L}_n$ contains only the next hops of the node.

F. Broadcast-Based Routing

In Broadcast-Based Routing (BBR), messages are flooded throughout the network, i.e., sent to all nodes whenever contacts are available. BBR does not require any knowledge about contacts or group membership in computing the NEXT-HOP list $\mathcal{L}_n$, which always includes all nodes in the network.

G. Unicast-Based Routing

In Unicast-Based Routing (UBR), multicast message transmission is implemented by sending multiple unicast messages. Specifically, when a multicast message is generated, the source node sends a unicast message, which encapsulates the original multicast message, to each of the estimated intended receivers. The source node also buffers the multicast message and sends out new unicast messages when being informed of new intended receivers. In this paper, we assume that unicast messages are forwarded using the shortest paths to the destinations. Unicast messages are removed from node storage after being transmitted to the next hop. Upon receiving a unicast message, the destination node will decapsulate the message and forward the original multicast message to the intended receiver according to the delivery constraints of the specified semantic model.

VI. Performance Evaluation

In this section, we evaluate the performance of the multicast routing algorithms presented in this paper using *ns* simulations. We aim to compare various routing algorithms and understand how the availability of knowledge about network topology and group membership affects routing performance. Our objective is to guide the design of multicast protocols for DTNs. In the following, we first describe our simulation methodology and performance metrics in Section VI-A and then present our results in the following sections.

A. Methodology and Metrics

In this paper, we simulate a specific type of DTNs, *sparse mobile networks* that consist of mobile nodes communicating via wireless radios. In these networks, nodes are sparsely distributed such that the networks experience frequent and long-duration partitions. Because of node movement, network topology changes over time.

We implement the four classes of routing algorithms and all the contact and membership oracles in the *ns* simulator. Our simulations use the IEEE 802.11 MAC layer. The radio range and data rate are 250m and 2Mbps respectively. To implement the complete contact oracle, we generate the movement of nodes for the entire simulation duration and then compute the DTN graph based on connectivity between nodes, i.e., whether nodes are within radio ranges of each other. Similarly we implement the contact summary oracle. Since the wireless medium is shared among nodes, the capacity of a contact may be affected by other nodes transmitting in vicinity, thus can not be estimated exactly in advance. In our simulations, we set the capacity of contacts as the radio data rate, i.e., 2Mbps, which is similar to the approximation model used in [21].

In our simulations, we use the following default settings unless specified otherwise. All simulations have 40 nodes on a $5000m \times 5000m$ area. Nodes move in the area according to the Random Way-Point (RWP) model [23] with a maximum speed 5m/s and a minimum speed 1m/s. The node storage capacity is 400 messages. In order to discover other nodes for communication, each node sends out beacon messages every 3 seconds. Each simulation lasts for 10000 seconds and each result is averaged over five runs with random seeds.

To understand how these routing algorithms perform, we consider only multicast traffic in the network. By default, there are 4 multicast sessions and each session consists of a single source which transmits messages to a multicast group. Each multicast group has 10 potential members which join and leave the group dynamically. Both the source and the potential group members are chosen randomly. Messages are generated at each source according to a Poisson process with mean inter-arrival time 4 seconds. Each message has 1000 bytes, thus the traffic rate of each source is 2kbps. After an endpoint joins (leaves) a group, it will leave (join) the group after a duration that is exponentially distributed with mean 200 seconds. Messages use the TD semantic model with membership interval $[t_0, t_0 + 100]$ and delivery interval $[t_0, t_0 + 3000]$ where t_0 is the message generation time, hence messages will be dropped after 3000 seconds.

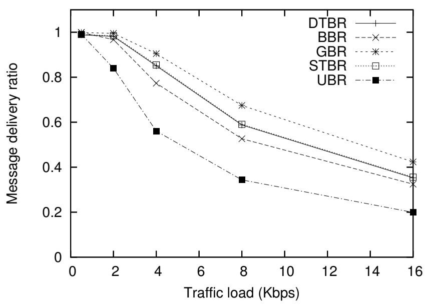
We evaluate the performance of multicast routing algorithms using the metrics defined in Section IV-A, namely the message delivery ratio, routing efficiency and delay.

B. Impact of Traffic Rate

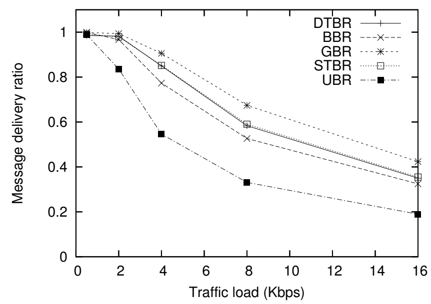
In this section, we study the impact of message generation rates on routing performance. The average message inter-arrival for all sources varies from 16, 4, 2, 1, to 0.5 seconds. Thus the total traffic load ranges from 0.5, 2, 4, 8, to 16 kbps. We first compare the performance between various algorithms. Fig. 7 shows the delivery ratio when different oracles are used. We make the following observations. First, the delivery ratio decreases for all algorithms as the traffic load increases, which is as expected. Second, among the routing algorithms that utilize knowledge in computing routes, GBR achieves the best performance. This is because in GBR, messages may be forwarded to receivers via multiple paths, which is better in exploiting available contact opportunities. UBR, on the other hand, has the worst delivery ratio because a separate unicast message is sent to each receiver which significantly increases contention for node storage and transmission opportunities, and results in message drops. This result confirms the intuition that providing multicast service by sending multiple unicast messages is very inefficient in DTNs. The performance of both DTBR and STBR is between that of GBR and UBR. Since DTBR can adapt to network conditions, it performs slightly better than STBR. We find that the relative performance among these algorithms remains the same in all simulations in this paper. This is consistent with the routing strategies these algorithms use. Third, the performance of BBR is independent of the available knowledge since it does not utilize knowledge in routing. Fourth, GBR and BBR achieve the highest delivery ratio, depending on the level of available knowledge. GBR performs best most of the time when complete contact knowledge is available, while BBR achieves the best delivery ratio when the contact summary oracle is used. Both BBR and GBR utilize some form of flooding in message forwarding, which suggests that forwarding messages via multiple paths is a promising approach to achieve high delivery ratio in DTNs.

We now study how each routing algorithm performs under different amount of available knowledge. Fig. 8 shows the results for GBR, which are representative of other algorithms that utilize knowledge, i.e., UBR, STBR and DTBR. The labels "CC-xM"("SC-xM") in the figure represent scenarios where the complete contact (contact summary) oracle is used. We can see that the availability of up-to-date membership and exact contact knowledge has significant effect on routing performance. GBR performs poorly when such knowledge is not available. This suggests that a minimum amount of knowledge is required to achieve efficient routing for these approaches. In addition, the marginal improvement in performance for accurate contact information is more significant than that for up-to-date membership information. We also find that the performance with the current membership oracle, which is not shown in the figure for clarity, is almost the same as that with the complete membership oracle.

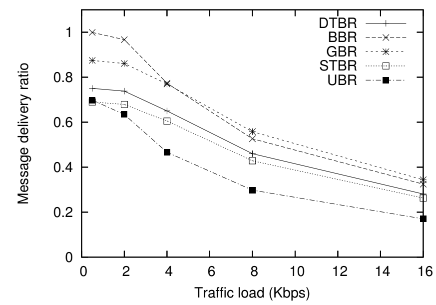
We also evaluate the routing efficiency and delay of various algorithms. Fig. 9(a) illustrates the routing efficiency of various algorithms when the complete contact and complete membership oracles are used. We can see that BBR, which uses flooding to forward messages, has the lowest routing efficiency because it generates many redundant messages. Thus BBR is not suitable for mobile networks where nodes are equipped with limited power supplies. UBR is also inefficient in utilizing resources since it sends a separate copy of a multicast message to every receiver. STBR and DTBR achieve the best routing efficiency among all algorithms. The routing efficiency for GBR is slightly lower than TBR algorithms. Fig. 9(b) depicts the routing efficiency when the contact summary and complete



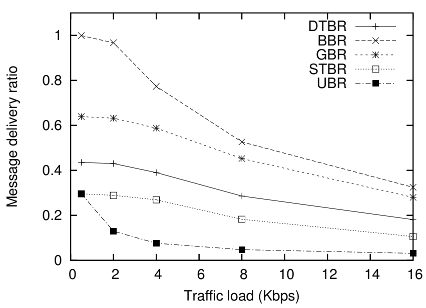
(a) Complete contact, Complete membership



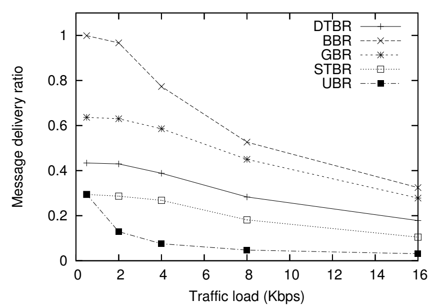
(b) Complete contact, Current membership



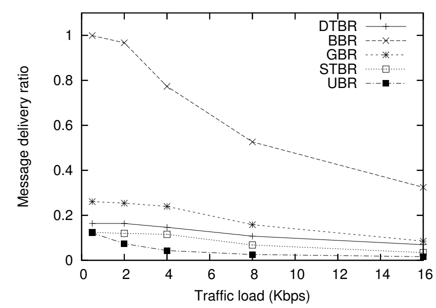
(c) Complete contact, Delayed membership



(d) Contact summary, Complete membership



(e) Contact summary, Current membership



(f) Contact summary, Delayed membership

Fig. 7. Message delivery ratio under different message generation rates.

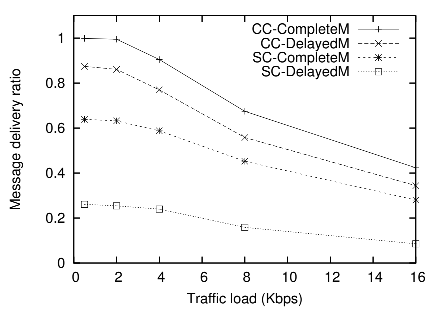


Fig. 8. Message delivery ratio of GBR under different knowledge.

membership oracles are used. We notice that GBR achieves better efficiency than both TBR algorithms in this case. In addition, UBR achieves the highest routing efficiency, which, however, is obtained with a very low delivery ratio.

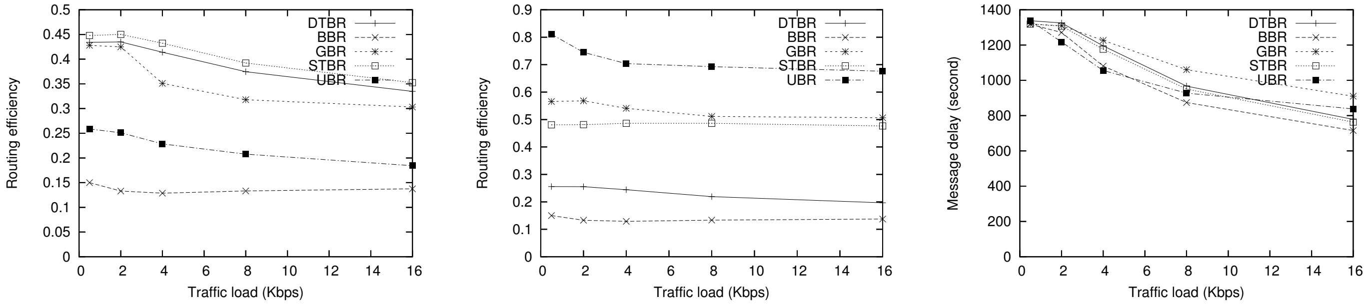
Fig. 9(c) shows the average delay for delivered messages when the complete contact and complete membership oracles are used. We can see that for all algorithms, the message delay decreases as the traffic load increases. This is because as the network becomes more congested, messages of the same age are more likely to be removed from node storage. Thus messages tend to reach only receivers that are on a shorter forwarding path, resulting in lower message delay. BBR achieves slightly lower delay than other algorithms because messages are flooded to all nodes and it is more likely that messages follow a shorter path to the receivers. Somewhat surprisingly, the delay of GBR is larger than that of STBR and DTBR.

C. Impact of Number of Sessions

In this section, we evaluate how the number of multicast sessions affects routing performance. In these simulations, each source generates messages every 16 seconds on average. The number of sources varies from 1, 4, 8, 16, to 32, so the total traffic load ranges from 0.5, 2, 4, 8, to 16 kbps. Fig. 10(a) shows the delivery ratio when the complete contact and complete membership oracles are used. We can see that the delivery ratio decreases as the number of sessions increases. This is as expected because the total traffic load in the network increases. The performance of BBR drops more rapidly than other algorithms since BBR generates more traffic in the network. So the contention for node storage and transmission bandwidth intensifies, leading to lower delivery ratios. In addition, the relative performance among GBR, DTBR, STBR and UBR remains the same as in the previous simulations, i.e., GBR is better than STBR and DTBR which in turn are better than UBR. Fig. 10(b) shows similar results when the contact summary and complete membership oracles are used. One notable difference is that BBR performs significantly better than other algorithms when the number of sessions is small. Fig. 10(c) compares the performance of GBR under different levels of knowledge and the results are similar to those in Fig. 8.

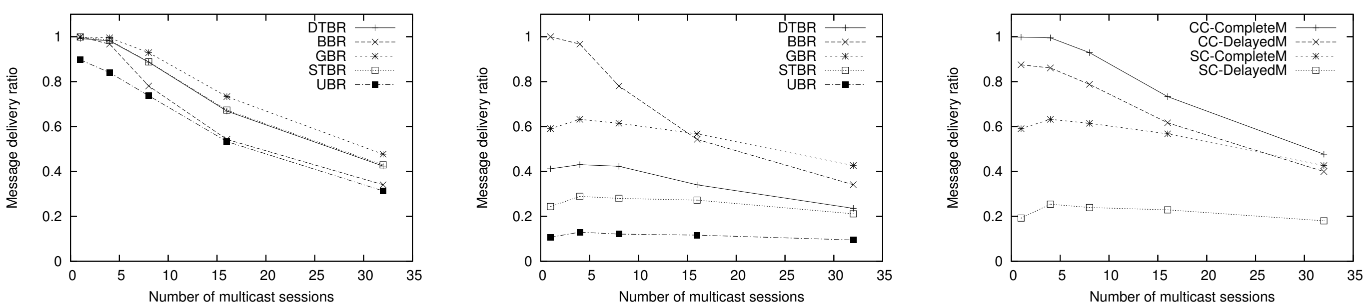
D. Impact of Session Size

In this section, we study how various routing algorithms perform under different session sizes, i.e., the number of potential group members in a multicast session. Since these potential members join and leave the group in the same fashion, the average number of group members over time is proportional to the session size. For a larger group, a message



(a) Complete contact, Complete membership (b) Contact summary, Complete membership (c) Complete contact, Complete membership

Fig. 9. Routing performance under different message generation rates.



(a) Complete contact, Complete membership (b) Contact summary, Complete membership (c) Delivery ratio

Fig. 10. Message delivery ratio for different numbers of multicast sessions.

needs to be delivered to more receivers, resulting in more traffic generated in the network.

Fig. 11(a) depicts the delivery ratio when the complete contact and complete membership oracles are used. We can see that the performance of BBR remains the same regardless of the session size. This is because BBR uses flooding to forward messages. The delivery ratios of GBR and both TBR algorithms decrease as the session size increases, a result of more generated traffic in the network. For UBR, the delivery ratio drops significantly as the session size increases because more unicast messages are generated. This confirms that UBR is not scalable for supporting large multicast groups. Fig. 11(b) shows the results when the contact summary and complete membership oracles are used. Under this scenario, the performance of GBR improves as the session size increases. The reason is that for a small session, the size of the forwarding group or the number of next hops to transmit a message is small. Due to the inaccuracy of contact information, GBR has a low delivery ratio. As the session size increases, GBR can transmit a message to more nodes as the next hop. Thus the limitation of partial knowledge becomes less significant. Fig. 11(c) compares the performance of GBR under different levels of knowledge. We can see that for small sessions, the performance of GBR varies significantly under different levels of knowledge. As the session size increases, however, the difference in performance becomes less notable except for the case with the least amount of knowledge (i.e., when the contact summary and delayed membership oracles are used).

E. Impact of Node Mobility

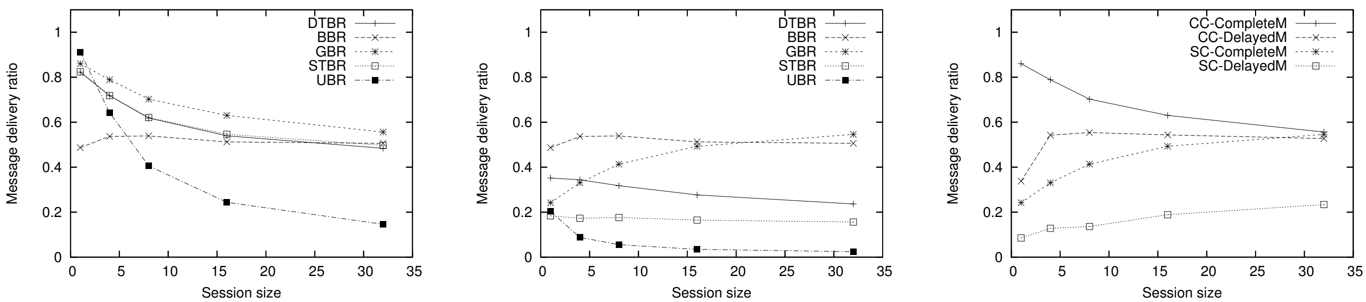
In this section, we study the impact of node mobility on routing performance. We consider both the Random Way-Point model and the Area-based Random Way-Point (ARWP) model. In ARWP, nodes move in the same way as in RWP except that their movement is confined in an area of size $200m \times 200m$ that is centered at a randomly chosen location. ARWP represents scenarios where nodes only move locally. In addition, we simulate two mobility levels. In the high level, the maximum node speed is 10m/s while in the low level, the maximum node speed is 5m/s. In both cases, the minimum speed is 1m/s.

Fig. 12 shows the delivery ratio when the complete contact and complete membership oracles are used. We can see that with the high mobility level, the delivery ratio is similar for all algorithms. This is because contact opportunities between nodes are ample when nodes move at a high speed. So messages can be delivered to the receivers even the algorithm chooses an inferior path. This suggests that when connectivity is ample, the choice of algorithms is of less importance.

However, the same is not true for the low mobility level. In these situations, contact opportunities between nodes are limited, thus route selection becomes important. We observe different performance among these algorithms. Similar to the previous simulations, GBR achieves the highest delivery ratio while UBR has the worst performance.

F. Impact of Node Storage Capacity

We now study the impact of node storage capacity on performance. Fig. 13 shows the delivery ratio when the complete



(a) Complete contact, Complete membership

(b) Contact summary, Complete membership

(c) Delivery ratio

Fig. 11. Message delivery ratio under different session sizes.

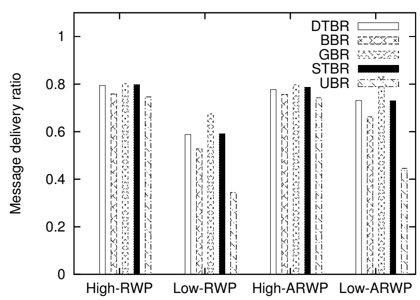


Fig. 12. Message delivery ratio under different mobility scenarios.

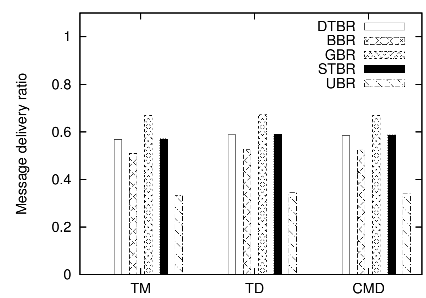


Fig. 14. Message delivery ratio under different multicast semantic models.

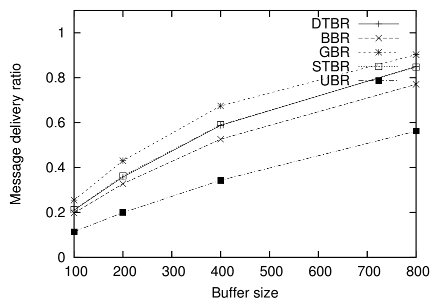


Fig. 13. Message delivery ratio under different buffer sizes.

contact and complete membership oracles are used. We can see that as the buffer size increases, the delivery ratios for all algorithms also increase. This is because in DTNs, messages are buffered in node storage for a long period of time. With more buffers, nodes would be able to drop fewer messages, leading to better delivery ratios. We obtain similar results when different oracles are used. This suggests that in DTNs, node storage has significant effect on routing performance.

G. Impact of Multicast Semantic Model

We now study the impact of multicast semantic models. When the TM model is used, there is no expiration time for messages. That is, messages are dropped only when buffer overflows. In the case of the TD and CMD models, the delivery constraint specifies that messages should be delivered within 3000 seconds. Fig. 14 shows the delivery ratio under different semantic models when the traffic rate for each source is 2kbps and the complete contact and complete membership oracles are used. When the CMD model is used, we assume that the CMD flags of messages are set. We can see that for all algorithms, the delivery ratio is similar under different semantic models. This is because under the relatively high traffic load in this scenario, messages will be dropped early due to buffer shortage. So the performance with the TM model is similar to that with the TD or CMD model. Fig. 14 also shows that the use of different semantic models does not affect the relative performance between various algorithms that utilize knowledge to compute routes, e.g., GBR performs best while UBR is the worst. We obtain results for the case when the traffic rate for each source is relatively low (0.25kbps). The delivery ratio for the TM model is higher than that of the TD or CMD model when the contact summary oracle is used because there is no message expiration in the TM model.

VII. Related Work

In this section, we review some related work on DTNs and multicasting in traditional networks. DTNs are assumed to experience frequent and long-duration partitions. This is in contrast to the traditional network model that assumes networks are connected. Examples of DTNs include military ad hoc networks [1], deep space communication [3], [9] and vehicular communication [35]. To achieve interoperability between various types of DTNs, Fall [17] proposes an architecture that is based on an asynchronous message forwarding paradigm. This architecture operates as an overlay above the transport layers to connect different DTNs.

Routing in frequent-disconnected networks has been studied relatively recently. In [21], Jain et al. study unicast routing in general DTNs and develop several routing algorithms for scenarios where different levels of knowledge about network is available. The authors present a framework to evaluate these algorithms and find that efficient routing can be achieved using only limited amount of knowledge. There is also other work that focuses on sparse mobile networks and exploits node mobility to deliver data. For example, Vahdat and Becker [32] propose Epidemic Routing in which mobile nodes carry data and exchange data when they meet, essentially flooding data throughout the network. In the Data Mules project, Shah et al. [29] propose to exploit mobile entities to transport data from sensors to access points, thus conserving energy in resource-limited sensors. In the Message Ferrying project, Zhao et al. [36], [37] propose the use of special nodes called *message ferries* to provide communication services and exploiting controlled node mobility to improve routing performance. Other work includes [6], [10], [11], [19], [24], [31], [33]. In all these studies, routing is achieved in a "store-carry-and-forward" fashion to overcome disconnection.

Multicasting has been studied extensively in the past, both in the Internet and in MANETs. Deering and Cheriton [15] first introduce the concept of IP multicasting. IP multicast assumes an open group model in which sources do not need to know the group membership or be group members to send data to a group. In addition, nodes can join or leave a multicast group at will. Various multicast protocols have been developed for the Internet, including DVMRP, MOSPF, PIM and CBT [5], [15], [16], [26] (see [4] for a survey of IP multicasting). These protocols construct a multicast tree to forward packets, using either a broad-and-prune (dense mode) or an explicit join (sparse mode) mechanism. Another multicast routing approach uses multi-destination addressing (or explicit addressing) [2], [7], [13] in which sources maintain group membership and multicast packets carry the unicast addresses of group members. Because of the overhead of carrying group member addresses, this approach is more suitable for small-group applications.

In MANETs, node mobility introduces frequent topological changes which is different from the wired Internet. In addition, MANETs are resource-constrained in terms of bandwidth and energy supplies, thus routing protocols in these networks should be efficient in resource usage. Due to these issues, multicast routing protocols that are designed for the Internet, where topology changes are rare, can not be directly adopted for MANETs. Many multicast protocols have been proposed for MANETs including AMRIS, CAMP, FGMP, MAODV, ODMRP, DDM, AMRoute and MCEDAR [8], [12], [18], [22], [25], [28], [30], [34] (see [14] for a survey of MANET multicasting). These protocols use various techniques to address the issues of node mobility and resource constraints, e.g., reactive (on-demand) routing to conserve energy, localized repair of broken paths, and the use of a mesh structure instead of a forwarding tree to avoid frequent reconfiguration in the present of node mobility.

In this paper, we study multicasting in DTNs. The semantic models we developed are based on the open group model used in IP multicasting. Due to membership changes during message transfer, additional temporal constraint is needed to identify receivers of a message, which is different from the IP multicast model. Multicast routing in DTNs shares some commonalities with that in MANETs, e.g., dynamic topology and limited resources. Our simulation results show that group-based routing achieves better performance than tree-based routing, which has been observed in previous research in MANET multicast routing. On the other hand, DTNs differ from MANETs in the following aspects, namely frequent partitions, large message delivery delay, message-based transmission, and highly delayed join/leave requests. These factors affects not only data forwarding, but also the dissemination of control information. It would be interesting to study how MANET routing protocols or techniques can be adapted to DTNs.

In [20], Huang et al. propose a new class of multicast called *mobicast* for sensor networks. In mobicast, applications can specify spatiotemporal constraints on a mobile delivery zone for a packet. In contrast, the multicast models proposed in this paper define the intended receivers of messages, which are time-invariant, by specifying temporal constraints on group membership, instead of geographic regions that change over time.

VIII. Conclusion

In this paper, we studied the problem of multicasting in DTNs. We focused on the multicast semantics and routing algorithms. In DTNs, due to frequent partitions and consequently large transfer delays, group membership changes during data transfer are the norm rather than the exception. Under these situations, it is necessary to distinguish group members and the intended receivers of a message. We developed three multicast semantic models that allow users to explicitly specify temporal constraints on group membership and message delivery. These semantic models unambiguously define the intended receivers of messages and have various applications in DTN environments.

With these semantic models, we developed four classes of routing algorithms for DTNs with different routing strategies, i.e., routing using multiple unicast messages, flooding, routing using a forwarding tree, and routing using a forwarding group. To evaluate these algorithms, we extended the evaluation framework in [21] for multicast routing. With extensive *ns* simulations, we compared these multicast algorithms and studied how routing performance is affected by the availability of knowledge.

Based on our simulations, we obtained the following results. First, with the least amount of knowledge, i.e., using the contact summary and delayed membership oracles, algorithms which utilize knowledge in computing routes perform poorly. However, with either accurate contact information or up-to-date membership information, these algorithms achieve significantly better performance. Thus, as in the case of unicast [21], efficient routing for multicast can be constructed using only partial knowledge. In addition, the marginal improvement in performance for accurate contact information is generally more significant than that for up-to-date membership information. Second, GBR and BBR achieve the best delivery ratios depending on the amount of knowledge available. Both algorithms use some form of flooding, which suggest that forwarding messages along multiple paths is a promising approach for multicasting in DTNs. Third, UBR performs poorly in DTNs, confirming that multicast routing using multiple unicast messages is not efficient in DTNs.

In this paper, we used the concept of knowledge oracles to study the impact of available knowledge on routing performance. The use of knowledge oracles does not consider the overhead in disseminating such knowledge in the network, which might affect routing performance. To address this problem, we are currently studying the problem of information dissemination in DTNs. We also plan to develop multicast routing protocols for DTNs based on the semantic models and routing algorithms presented in this paper. One possible approach is to adapt MANET routing protocols such as ODMRP to DTNs. In addition, we are interested in extending our multicast semantic models to incorporate spatial constraints as in geocast or mobicast.

References

[1] DARPA Disruption Tolerant Networking Program: http://www.darpa.mil/ato/solicit/dtn.

[2] L. Aguilar. Datagram routing for internet multicasting. In *ACM SIGCOMM'84*, 1984.

[3] I. F. Akyildiz, O. B. Akan, C. Chen, J. Fang, and W. Su. Interplanetary internet: State-of-the-art and research challenges. *Computer Networks Journal (Elsevier)*, 43(2), October 2003.

[4] K. Almeroth. The evolution of multicast: From the mbone to interdomain multicast to internet2 deployment. *IEEE Network*, January/February 2000.

[5] T. Ballardie, P. Francis, and J. Crowcroft. Core Based Trees (CBT): An architecture for scalable inter-domain multicast routing. In *ACM SIGCOMM*, Sept. 1993.

[6] A. Beaufour, M. Leopold, and P. Bonnet. Smart-tag based data dissemination. In *First ACM WSNA Workshop*, September 2002.

[7] R. Boivie, N. Feldman, Y. Imai, W. Livens, D. Ooms, O. Paridaens, and E. Muramoto. Explicit multicast (xcast) basic specification. Internet-Drafts, Jan. 2005.

[8] E. Bommaiah, M. Liu, A. McAuley, and R. Talpade. AMRoute: Ad hoc multicast routing protocol. *Internet draft*, Aug. 1998.

[9] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss. Delay-tolerant networking – an approach to interplanetary internet. *IEEE Communications Magazine*, June 2003.

[10] B. Burns, O. Brock, and B. N. Levine. MV routing and capacity building in disruption tolerant networks. In *IEEE Infocom 2005*, March 2005.

[11] A. Chakrabarti, A. Sabharwal, and B. Aazhang. Using predictable observer mobility for power efficient design of sensor networks. In *Information Processing in Sensor Networks*, Palo Alto, CA, April 2003.

[12] C.-C. Chiang, M. Gerla, and L. Zhang. Forwarding group multicast protocol (FGMP) for multihop, mobile wireless networks. *Cluster Computing*, 1(2):187–196, 1998.

[13] Y. K. Dalal and R. M. Metcalfe. Reverse path forwarding of broadcast packets. *Commun. ACM*, 21(12), 1978.

[14] C. de Morais Cordeiro, H. Gossain, and D. P. Agrawal. Multicast over wireless mobile ad hoc networks: Present and future directions. *IEEE Network*, 17(1), Jan 2003.

[15] S. Deering and D. Cheriton. Multicast routing in datagram internetworks and extended LANs. *ACM Transactions on Computer Systems (TOCS)*, 1990.

[16] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu, and L. Wei. An architecture for wide-area multicast routing. In *ACM SIGCOMM 1994*, 1994.

[17] K. Fall. A delay-tolerant network architecture for challenged internets. In *ACM Sigcomm 2003*, 2003.

[18] J. J. Garcia-Luna-Aceves and E. L. Madruga. The core-assisted mesh protocol. *IEEE J. Select. Areas Commun.*, 17:1380–1394, Aug. 1999.

[19] A. A. Hasson, R. Fletcher, and A. Pentland. DakNet: A road to universal broadband connectivity. *Wireless Internet UN ICT Conference Case Study*, 2003.

[20] Q. Huang, C. Lu, and G.-C. Roman. Spatiotemporal multicast in sensor networks. In *ACM SenSys 2003*, pages 205–217, 2003.

[21] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. In *ACM Sigcomm 2004*, Portland, OR, 2004.

[22] L. Ji and M. Corson. Differential destination multicast – a MANET multicast routing for multihop, ad hoc network. In *IEEE INFOCOM*, 2001 Apr.

[23] D. Johnson and D. Maltz. Dynamic source routing in ad-hoc wireless networks. In *ACM SIGCOMM*, August 1996.

[24] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet. In *ASPLOS'02*, October 2002.

[25] S. Lee, M. Gerla, and C. Chiang. On-demand multicast routing protocol (ODMRP) for ad hoc networks. *Internet draft*, June 1999.

[26] J. Moy. Multicast extensions to OSPF. *RFC 1584*, Mar. 1994.

[27] Network simulator 2. http://www.isi.edu/nsnam/ns/.

[28] E. Royer and C. Perkins. Multicast operation of the ad-hoc on-demand distance vector routing protocol. In *ACM Mobicom*, Seattle, WA, Aug. 1999.

[29] R. Shah, S. Roy, S. Jain, and W. Brunette. Data MULEs: Modeling a three-tier architecture for sparse sensor networks. In *IEEE SNPA Workshop*, 2003.

[30] P. Sinha, R. Sivakumar, and V. Bharghavan. MCEDAR: multicast core-extraction distributed ad hoc routing. In *IEEE WCNC*, Sept. 1999.

[31] Tara Small and Zygmunt Haas. The Shared Wireless Infostation Model - A New Ad Hoc Networking Paradigm (or Where there is a Whale, there is a Way). In *ACM MobiHoc*, June 2003.

[32] A. Vahdat and D. Becker. Epidemic routing for partially-connected ad hoc networks. *Technical report, Duke University*, 2000.

[33] R. Wang, S. Sobti, N. Garg, E. Ziskind, J. Lai, and A. Krishnamurthy. Turning the postal system into a generic digital communication mechanism. In *ACM SIGCOMM 2004*, August 2004.

[34] C. Wu, Y. Tay, and C.-K. Toh. Ad hoc multicast routing protocol utilizing increasing id-numbers (AMRIS) functional specification. *Internet draft*, Nov. 1998.

[35] H. Wu, R. Fujimoto, and G. Riley. Analytical models for data dissemination in vehicle-to-vehicle networks. In *IEEE VTC 2004/Fall*.

[36] W. Zhao and M. Ammar. Proactive routing in highly-partitioned wireless ad hoc networks. In *the 9th IEEE International Workshop on Future Trends of Distributed Computing Systems*, May, 2003.

[37] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *ACM MobiHoc 2004*, Tokyo Japan, 2004.