Performance Comparison of Different Multicast Routing Strategies in Disruption Tolerant Networks

Qing Ye, Liang Cheng, Mooi Choo Chuah, and Brian D. Davison
Department of Computer Science and Engineering, Lehigh University
Bethlehem, PA 18015, USA
{qiy3, cheng, chuah, davison}@cse.lehigh.edu

***Abstract*—Disruption Tolerant Networks (DTNs) technologies are emerging solutions to networks that experience frequent partitions. As a result, multicast design for DTNs is a considerably more difficult problem compared to that of Internet and mobile ad hoc networks. In this paper, we first investigate three basic DTN multicast strategies, including unicast-based multicast (U-Multicast), static-tree-based multicast (ST-Multicast) and dynamic-tree-based multicast (DT-Multicast). Then we focus on studying two DT-Multicast methods: Dynamic Tree Based Routing (DTBR) and On-demand Situation-aware Multicast (OS-Multicast), which address the challenges of utilizing opportunistic connectivity to conduct one-to-many data communication in DTNs. Performances of different multicast approaches are evaluated by simulations. Our results show that DT-Multicast approaches can achieve higher message delivery ratio than the other strategies. Also, to get better performance, we recommend that system designers select OS-Multicast when the traffic load is low and select DTBR when the traffic load is high respectively.**

Keywords-disruption/delay tolerant networks; intermittently connected networks; dynamic multicasting; performance evaluation

I. Introduction

Recent research in Disruption Tolerant Networks (DTNs) [1-4, 28-30] addresses challenges in handling data communications in networks where instantaneous end-to-end paths become unstable and inconsistent. Communication environments of DTNs feature frequent network partitions as the result of planned or unplanned link up/down periods between neighboring nodes. Such link layer challenges may be caused by high node mobility, low network density, limited radio ranges, scheduled node unavailability, or unexpected infrastructure disruptions. DTN technologies are viewed as building components to support a broad range of applications such as military battlefield surveillance [5], deep-space communications [6], and Internet access in rural areas [7].

A. Motivations

DTN multicast is an important service to deliver messages from the sources to a group of receivers in the disruption tolerant networks. Network designers face some challenges when directly applying multicast routing methods proposed for the Internet (e.g., MOSPF [8] and DVMRP [9]) or mobile ad hoc networks (e.g., AMRoute [10] and ODMRP [11]) to DTN environments. First, it is difficult to maintain the connectivity of a multicast structure (tree or mesh) during the lifetime of a multicast session. Second, data transmissions would suffer from many failures and large end-to-end delays because of the disruptions caused by repeatedly broken multicast branches. Third, the traditional approaches are designed with the assumption that the underlying networks are basically connected, which is not true in most DTN environments [30].

To address these issues, several DTN multicast strategies have been proposed, including unicast-based, static-tree-based and dynamic-tree-based multicast approaches. Each strategy has its own advantages and drawbacks due to different design methodologies. Within each strategy, one can design different variations of multicast routing schemes. The major goal of this paper is to study the performances of these multicast strategies in different DTN scenarios.

Our performance metrics include: *i*) *message delivery ratio*, which is defined as the number of unique multicast messages successfully delivered to the receivers over the total number of messages that are expected to be received; *ii*) *message delivery efficiency*, which is the ratio between the unique messages received by the receivers and the total traffic generated in the network; and *iii*) *average message delay*, which is the average of end-to-end message transmission delays. From our investigations and discussions, we aim to help system designers select the appropriate multicast routing scheme to meet the design requirements such as to achieve the most message delivery ratio, the smallest delay, or the best efficiency.

B. Network Model

DTNs are viewed as an overlay built upon certain underlying networks, such as mobile ad hoc networks as illustrated in Fig. 1. Only those nodes that implement DTN functionalities, e.g. the support of sending and receiving bundles, are considered as DTN nodes, while the others are denoted as normal nodes. This overlay that consists of all the DTN nodes is named as the DTN layer. It is also called the bundle layer in [22]. Our discussions in the rest of this paper only focus on the DTN layer.

A DTN nodes N_j is called a *neighbor* of N_i if currently there is one end-to-end path connecting them in the underlying network. We use link e_{ij} to represent the current connectivity from node N_i to N_j in the DTN layer. The state of e_{ij} is *up* if and only if N_j is a neighbor of N_i. And the status of e_{ij} becomes *down* when there is no paths in the underlying networks connecting N_i to N_j at present. It is *up* again if at least one old path is reconnected or a new path is discovered. DTN unicast routing schemes are utilized in the DTN layer to deliver a bundle from one DTN node to one of its DTN neighbors. When a DTN node has bundles destined for a neighbor and there is currently no available outgoing link to reach the destination, it will retain the data in its local buffer. We assume that each DTN node has a finite-size buffer for storing bundles.

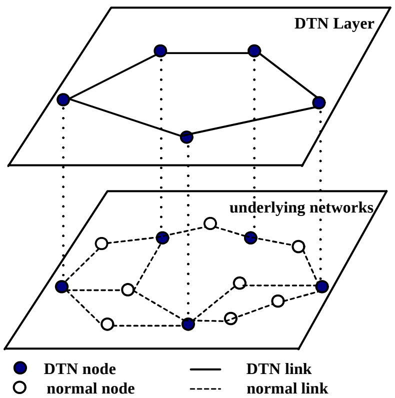


Figure 1. An example of the DTN layer.

C. *Problem Definition*

To define the multicast problem, we have the following notations:

- V, the set of nodes in the DTN layer. $V = \{N_1, N_2,\ldots,N_n\}$, where N_i represents the ith node and $|V|$=n.
- $E(t)$, the set of DTN links at time t between nodes in V. $e_{ij}(t) \in E(t)$ represents the directed link from N_i to N_j at time t.
- R, the set of the intended receivers. $R \subset V$ and $1<|R|<$n.
- S, the set of the multicast sources that hold messages for all the receivers in R. $S \subset V$, $SR=\phi$ and $0<|S|<$n-1 (there are at least 1 source and 2 receivers).
- D, the set of data that the sources intend to deliver to all the receivers in R.
- $G(t) = (V, E(t))$, a directed graph which represents the time varying DTN layer at time t.
- T_s, the start time of the multicast session.
- T_e, the end time of the multicast session. $T_e - T_s = L$.
- C, the maximum number of bundles that can be hold in the local buffer within each DTN node.

The DTN multicast problem is defined as: given a time variant network $G(t) = (V, E(t))$, how to find a set of multicast structure $MS(t_0)$, $MS(t_1)$,…, $MS(t_L)$ which are the sub-graphs of $G(t_0)$, $G(t_1)$,…, $G(t_L)$ where $T_s=t_0<t_1<\ldots<t_L=T_e$, to route the data in D from S to the receivers in R, under the constraints of the link connectivity and local storage availability in $G(t)$. Those multicast structures could be either tree or mesh which is not guaranteed to be connected in the DTN layer.

The rest of this paper is organized as follows. We introduce the ideas of different DTN multicast strategies including: U-Multicast, ST-Multicast and DT-Multicast in Section II. Section III explains the details of the DT-Multicast strategy by illustrating two DT-Multicast routing schemes. Performance comparisons are conducted and presented in Section IV. Section V shows the related works. Our conclusions are detailed in Section VI.

II. DTN MULTICAST STRATEGIES

In this section, we introduce the ideas of three DTN Multicast strategies. An example of them is shown in Fig.2.

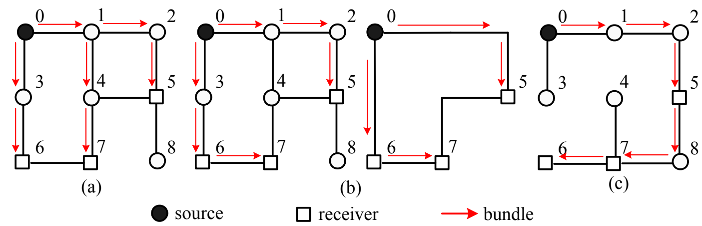


Figure 11. Three DTN multicast strategies. Assume that node 0 is the source and nodes 5,6,7 are the receivers. (a) U-Multicast strategy. (b) ST-Multicast strategy: multicast tree is created from the multicast overlay that only includes the multicast source and receivers. (c) DT-Multicast strategy: when links 3→6, 1→4, 4→5 are down, but link 8→7 is up, a new multicast tree is dynamically built to take advantage of the newly available link 8→7 to forward bundles to the destination.

A. *Unicast-based Multicast (U-Multicast) Strategy*

The simplest way to perform one-to-many data communication is to send the bundles via multiple unicast operations from the source to each destination. Any DTN unicast routing scheme discussed in Section V can be applied and extended to perform this task, with adding additional group information in the header of bundles. Some unicast routing schemes, such as the Epidemic Routing [18] and Spray-and-Wait [20] algorithms, already distribute the same bundles to multiple recipients. Therefore, U-Multicast strategy has the obvious advantage of the least implementation overheads. However, for $|R|$ numbers of receivers registered in a multicast group, U-Multicast requires the source deliver $|R|$ copies of each bundle to these receivers. Thus, the intermediate DTN nodes may repeatedly forward the same copy more than once. It then incurs the transmission overheads which may dramatically lower the message delivery efficiency of U-Multicast when there are many receivers.

B. *Static-tree-based Multicast (ST-Multicast) Strategy*

A spanning tree or a steiner tree is the typical structures which is widely used by many multicast protocols. The usage of a tree has been proven to be capable of reducing the transmission overheads in mobile ad hoc networks (MANET) [27]. In ST-Multicast strategy, a tree is constructed and maintained at the source when a multicast session starts. The source first collects all the discovered paths to the receivers and then builds up a smallest cost tree based on such information. By definition, the topology of the tree does not change in the intermediate nodes during the multicast session. And bundles are duplicated at every branching node based on how many downstream neighbors the node has. The overlay multicast approach proposed in [32] in MANET can be extended to be a ST-Multicast routing scheme in DTNs. It creates an additional multicast overlay which includes all the multicast group members in the DTN layer. Then the tree is built up based on the logical topology in this virtual overlay no matter how the topology in the DTN layer varies. ST-Multicast strategy is appropriate for some scenarios in which the network disruptions periodically happen in a fixed/scheduled pattern, for instance, the data communication via LEO satellites. In these cases, the discovered tree can act as the multicast backbone for delivering bundles. ST-Multicast strategy has smaller control overheads in terms of maintaining the multicast states than DT-Multicast strategy discussed next. But it loses the flexibility of adjusting the multicast routing decisions according to the topology variations in DTNs.

C. Dynamic-tree-based Multicast (DT-Multicast) Strategy

DT-Multicast strategy dynamically adjusts the multicast tree to adapt to the current conditions in the networks. In DT-Multicast, each bundle has an associated tree that may change hop-by-hop according to the up/down variations of DTN links. Each node that has a bundle performs the same operations: *i*) to collect the information of the availability of DTN links to update its knowledge of the networks; *ii*) to compute the smallest cost tree based on its latest local view of the DTN layer; and *iii*) to forward bundles using the discovered multicast tree. In DT-Multicast strategy, a node is capable of changing the multicast structure to take advantage of a newly available path to a destination or to avoid forwarding bundles along those currently broken branches that are discovered in the previous tree. Compared to the other two strategies, DT-Multicast strategy has more implementation and control overheads but is better adaptive to the topology variations in DTNs.

III. Details of DT-Multicast Strategy

In this section, we discuss the details of the DT-Multicast strategy by illustrating two typical DT-Multicast routing schemes: dynamic tree based routing (DTBR) and on-demand situation-aware multicast (OS-Multicast).

A. Basic Ideas of DTBR & OS-Multicast

DTBR [12] assumes that each DTN node has a certain knowledge oracle containing the schedule or the statistical summary of link up/down information in the DTN layer. Based on this, the source computes a multicast tree for each bundle and forwards the current message along the tree. There is a receiver list associated with each copy of the bundle. It indicates for which receivers an intermediate node should be responsible. Initially, the list at the source contains all the intended receivers. If the source has more than one downstream node, it will put a new list that only consists of the receivers along that branch into the copy sent to each downstream next hop. Each node that receives a bundle will then re-compute a multicast tree to reach those destinations in the receiver list. This process is repeated hop-by-hop until a copy of the bundle is delivered to a receiver.

OS-Multicast [13] is similar to DTBR as it also builds up a dynamic multicast tree hop-by-hop for each copy of the bundle. However, it doesn’t rely on any global knowledge of the network, such as node positions or link up/down schedule. It assumes that the underlying network is able to record discovered routing information and report the current availability of outgoing links to the DTN multicast agent. In OS-Multicast, there is also a receiver list associated with each bundle. Unlike DTBR, it always contains a full list of all the intended receivers. Therefore, OS-Multicast requires each intermediate node that has a bundle be responsible for delivering the multicast message to all the receivers.

B. Common Operations of DTBR & OS-Multicast

1) Membership management

Each DTN node is assumed to be associated with an *endpoint ID*. A multicast source (or multiple sources) uses a group endpoint ID or an explicit list of the receivers as the destination address for delivering multicast bundles.

Several semantic models of DTN multicast membership have been studied in [12]. The DTN multicast strategies discussed above can use the Temporal Membership (TM) semantic model with an explicit receiver list known at the source. When a DTN node intends to join a multicast group, it registers its planned membership period by explicitly flooding a *GROUP_JOIN* message (e.g., node *i* claims to be interested in the multicast service during the period $[t_{si}, t_{ei}]$, with the start-time t_{si} and the end-time t_{ei}.) into the DTN layer. When the multicast source is informed by the *GROUP_JOIN* message, it puts the membership information into a *receiver_list*, denoted as L_M. For each bundle to be transmitted at time *t*, a DTN node will check the validity of receivers in L_M. If the membership of a receiver has expired ($t > t_{ei}$) or was not activated ($t < t_{si}$), then that receiver will not be included in the receiver list in the bundle. An intended receiver with expired membership will be removed from L_M. A new *GROUP_JOIN* message is required when this receiver wants to participate in the multicast service again in the future.

2) Multicast tree construction

As mentioned, DTBR and OS-Multicast collects the information of the DTN layer by either querying the knowledge oracle or gathering the discovered routing information from the underlying unicasting method. Both compute the multicast tree by the Dijkstra shortest path tree algorithm using the hop distance as the cost, at each DTN node involved in the multicast transmission. The multicast structure is calculated dynamically based on the local view of the DTN layer at each intermediate node. The generic pseudo codes of how DT-Multicast strategies make the multicast routing decisions and how the bundles are forwarded are shown in Fig. 3 and Fig.4.

```
Algorithm: DT-Multicast Decision
Input:   G(t) = (V, E(t)), bundle
output:  MSi(t), the multicast structure created by node Ni at time t,
         Actioni(bundle, t), how node Ni process bundle at time t

foreach receiver r in bundle.LM do
    if notvalidreceiver(r, t) then
        bundle.LM = bundle.LM - r
    end if
end
G'(t) = CollectLocalViewofDTNLayer(G(t), i)
Costi(t) = CalculateHopCounts(G'(t), i)
MSi(t) = Dijksta_ShortestPathTree(G'(t), Costi(t), i)
actioni(t) = 0
foreach receiver r in bundle.LP do
    if r is reachable from Ni in MSi(t) then
        bundle.LP = bundle.LP - r
        actioni(t) = 1
    end if
end
if actioni(t) = 1 and |bundle.LP| > 0 then
    actioni(t) = 2
end if
```

Figure 12. Pseudocode of Generic DT-Multicast Strategy

```
Algorithm: DT-Multicast Action
Input:   actioni(t), MSi(t), bundle
output:  how node Ni process bundle at time t

if actioni(t) != 1 then
    if |localbufferi(t)| < maxsize then
        localbufferi(t) = localbufferi(t) + bundle
    else
        managelocalstorage(policy, localbufferi(t), bundle)
    end if
end
if actioni(t) != 0 then
    nexthoplist = getthenexthopsfromtree(MSi(t))
    foreach nexthop d in nexthoplist
        newLM = calculatereceiverlist(bundle.LM, MSi(t))
        newbundle = createcopy(bundle)
        newbundle.LM = newLM
        newbundle.LP = newLM
        newbundle.LU = newbundle.LU + Ni
        forwardbundle(d, newbundle)
    end
end if
```

Figure 13. Pseudocode of storing and forwarding bundles using the DT-Multicast strategy

3) Multicast state maintainess

To dynamically maintain the tree, each bundle keeps a unique forwarding state, including an *upstream_list* (called L_U) and a *pending_list* (called L_P). The upstream_list L_U contains the endpoint ID of DTN nodes a bundle has traversed. The purpose of L_U is to avoid possible routing loops and to reduce redundant traffic. When a bundle arrives, a DTN node creates L_P for that bundle by duplicating the associated receiver list L_M contained in the bundle. So initially $L_P = L_M$. After generating the multicast tree based on current network situations, it knows which receivers are reachable using this tree. Those covered receivers are then removed from L_P. If the resulting L_P is not empty, this node will put a copy of that bundle into its local buffer and wait for the future opportunities to reach those destinations stored in L_P. Otherwise, the bundle doesn't need to be buffered since all the receivers have been covered from its local view of the DTN layer.

Each DTN node that has buffered bundles will periodically check if there is any chance to forward the buffered bundles further. If so, it then recalculates a multicast tree to reach the uncovered receivers in L_P for each stored bundle. If one receiver is reachable by this new dynamic tree, it would be removed from L_P, and a copy of that bundle will be forwarded. A buffered bundle is released when *i*) its lifetime has expired; *ii*) its associated L_P is empty; or *iii*) the current buffer management policy decides to discard this bundle to avoid possible buffer overflow. Fig. 5 shows the pseudocode of maintaining the multicast states of buffered bundles in DT-Multicast strategy.

```
Algorithm: DT-Multicast Maintaining Operations
Input:   MSi(t)
output:  how node Ni maintain the multicast state of buffered bundles

if |localbufferi(t)| > 0 then
    foreach buffered bundle b in localbufferi(t)
        foreach receiver r in b.LP
            if r isreachable from Ni by MSi(t) then
                d = getthenexthop(MSi(t), r)
                if d is not in bundle.LU then
                    b.LP = b.LP - r
                    newLM = calculatereceiverlist(b.LM, MSi(t))
                    newbundle = createcopy(b)
                    newbundle.LM = newLM
                    newbundle.LP = newLM
                    newbundle.LU = newbundle.LU + Ni
                    forwardbundle(d, newbundle)
                end if
            end if
        end
        if |b.LP| = 0 then
            localbufferi(t) = localbufferi(t) - b
        end if
end if
```

Figure 14. Pseudocode of Generic DT-Multicast Strategy

C. Differences between DTBR & OS-Multicast

Different DT-multicast routing schemes differ in how the current information of the DTN layer are collected. DTBR requires that each node has complete knowledge or a summary of the link states in the networks. However, this is difficult to satisfy in most practical applications. Without such a strong requirement, OS-multicast may operate over any DTN unicast method that is able to record historical routing information and detect the status of outgoing links.

Another significant difference between these two DT-Multicast methods is how they treat the receiver list L_M in multicast bundles. In DTBR, L_M will be divided into subsets at each branching node in the multicast tree, while in OS-Multicast L_M will not change as bundles are transmitted. This simple modification allows OS-Multicast to utilize some bundle forwarding opportunities which are ignored by DTBR at the cost of additional redundant transmissions.

Consider two receivers N_i and N_j, both belong to the receiver

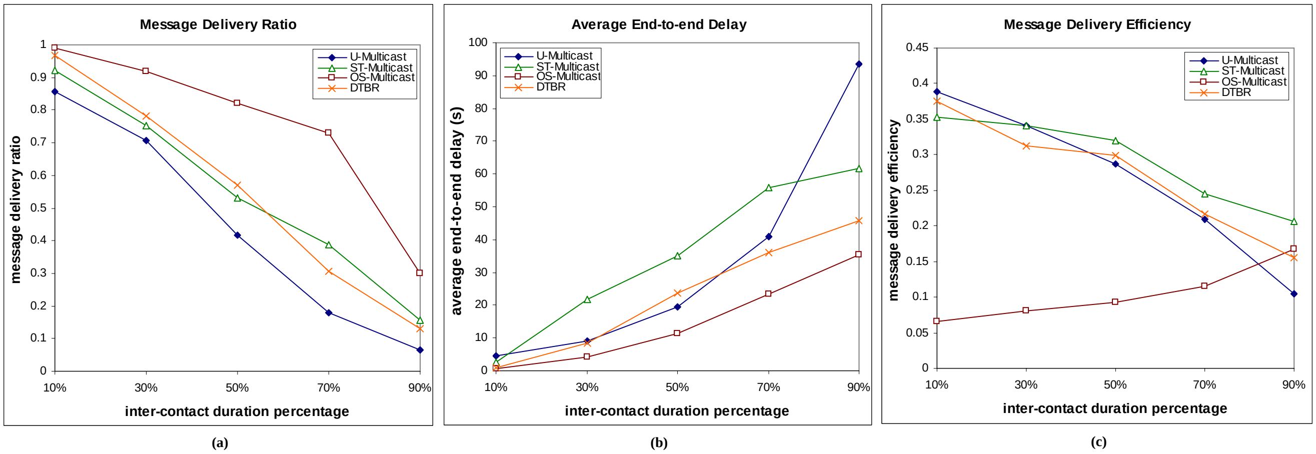


Figure 6. Simulation results of varying the inter-contact duration

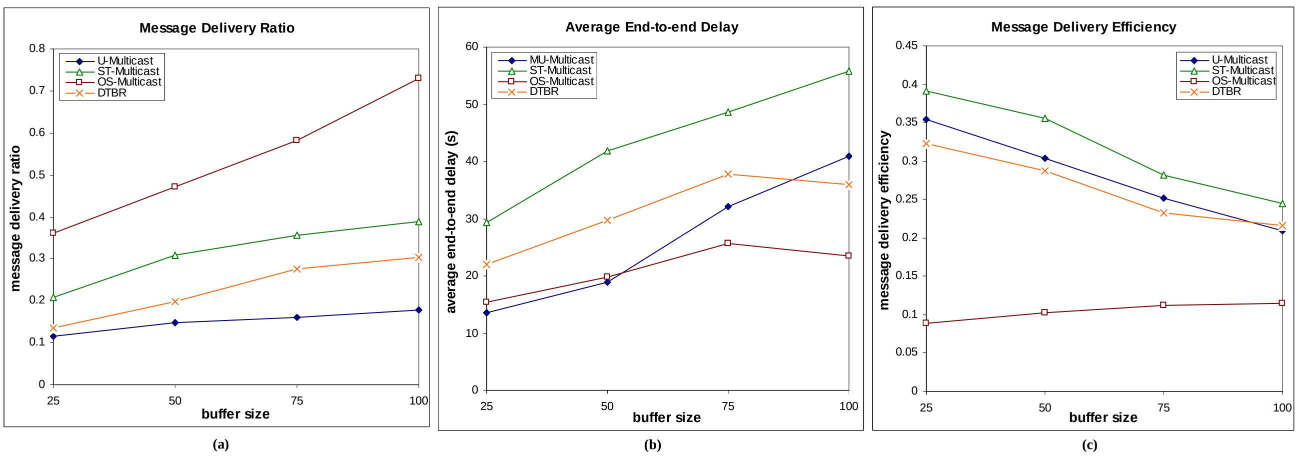


Figure 7. Simulation results of varying the local buffer size of each DTN node

set *R*. We denote $H_i(t_1)$ as the set of neighboring DTN nodes of N_i at time t_1. For node *X* in the set of $H_i(t_1) - H_i(t_1) \cap H_j(t_1)$, if it has a bundle then *X* would deliver the bundle to N_i only, according to DTBR. Suppose that at time $t_2 > t_1$, there occurs a link connecting *X* and N_j, i.e., $X \in H_j(t_2)$, as the result of the dynamic topology variation in DTNs. Then bundles held by node *X* at t_2 can not be forwarded to N_j immediately, because *X* is not required to be responsible for covering N_j before t_2. The opportunity provided by link $X \rightarrow N_j$ will not be utilized by DTBR until *X*'s upstream node in the tree discovers the new link and puts N_j into the receiver list of bundles forwarded to *X*. Moreover, bundle forwarding in DTBR stops at receivers N_i and N_j, if they are the leaf nodes of the multicast tree. Therefore, bundles successfully delivered to N_i will never be forwarded to N_j, even if there is currently an available link between N_i and N_j. Such issues with DTBR are overcome in OS-Multicast by keeping L_M the same along the tree.

However, OS-Multicast has the obvious disadvantage that it introduces much more redundant traffic into the network than DTBR. The performance of OS-Multicast will deteriorate when the traffic rate at the source is high, since redundant copies of delivered bundles will consume scarce wireless bandwidth. The performances of DTBR and OS-Multicast are detailed in Section IV.

IV. Performance Comparisions

To evaluate the performance of different multicast strategies, we implement the ideas of U-Multicast, ST-Multicast, DT-Multicast (including DTBR and OS-multicast) in the *ns-2* simulator [23].

A. *Results Without Node Mobility & Under Low Traffic Rate*

TABLE I. Simulation Parameters

Parameter	Value
Number of nodes	25
Area size	1,000m × 1,000m
Topology	5×5 grid
Number of multicast sources	1
Number of multicast receivers	4
Bundle size	512 bytes
Source traffic rate	1 bundle/2 seconds
Bundle retransmission interval	5 seconds
MAC layer protocol	802.11
Transmission range	250m
Simulation length	1,000 seconds

The first set of simulations is conducted under the condition of a low source traffic rate (1 bundle per 2 seconds) at the

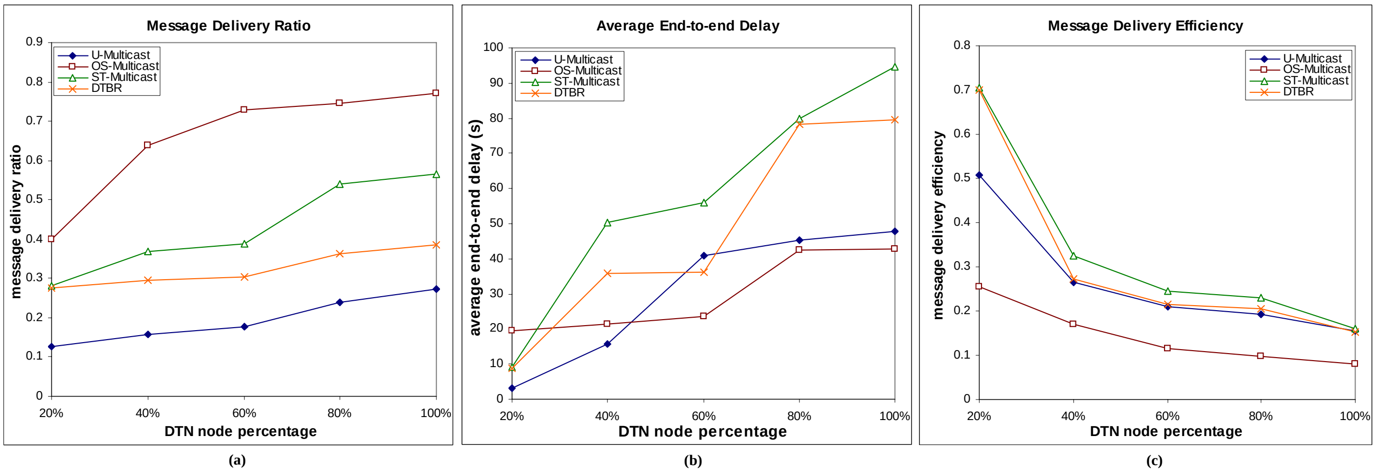


Figure 8. Simulation results of varying the scale of the DTN layer

multicast source and no node mobility. The purpose is to have a basic overview of the performance of different multicast strategies. The experiments are conducted by varying the inter-contact duration (defined in Section V.B.1), the local buffer size within DTN nodes, and the scale of the DTN layer. We also apply real-world DTN traces in the simulations. The common parameters of these simulations are listed in Table I. To emulate the characteristics of disruption tolerant networks, we have modified the ns-2 simulator to manage the link up/down schedules.

1) The impact of the inter-contact duration

In this simulation, each link experiences a series of randomly generated up/down periods. The inter-contact duration is the down time between two up time periods. For instance, if the up time of a link spans $[t_1, t_2]$, $[t_3, t_4]$, and $[t_5, t_6]$, then $[0, t_1]$, $[t_2, t_3]$, and $[t_4, t_5]$ are the inter-contact durations of this link. In this test, we vary the total length of the inter-contact durations of each link from 10% to 90% of the overall simulation time randomly. 15 out of 25 nodes are selected to be DTN nodes. And each DTN node can keep at most 100 bundles in its local buffer.

Fig. 6 shows the results. We observe that: *i*) fewer bundles could be delivered to the destinations when the percentage of inter-contact duration becomes larger, i.e., the lack of network connectivity becomes more severe; and *ii*) OS-Multicast achieves the best message delivery ratio because of its dynamic nature plus flooding to utilize almost all the opportunistic links to forward data to the destinations. Fig. 6(b) depicts the average end-to-end delay performance. Both U-Multicast and ST-Multicast perform worse than two DT-Multicast approaches. The reason is that they lack the flexibility of dynamically adjusting the multicast structure according to the changes in the networks so that both miss a lot of data forwarding opportunities and bundles suffer larger queuing delays. OS-Multicast achieves the smallest delay. However, OS-Mulicast has the worst message delivery efficiency when the percentage of inter-contact duration is less than 75% as shown in Fig. 6(c). Because each intermediate node in the multicast structure of OS-Multicast keeps the whole list of receivers, multiple copies of the same bundle may be delivered when the network remains well connected. But, as the percentage of inter-contact duration becomes small, the efficiency of OS-Multicast will benefit from its capability to deliver more bundles than the other three algorithms.

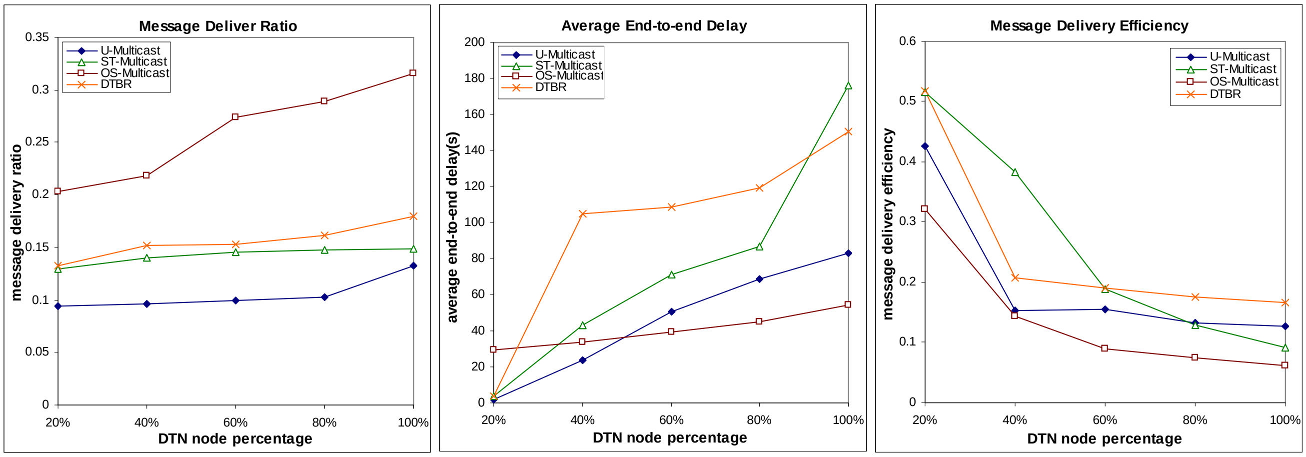
2) The impact of local buffer size

The size of local storage within each DTN node affects the performance of different multicast strategies. If the size is too small, some bundles have to be discarded due to buffer overflow. Those bundles then immediately lose the chance to be forwarded even when some opportunistic links would be available in the near future.

In this simulation, we fix the percentage of the inter-contact duration to be 70% of the total simulation time and vary the buffer size of each DTN node from holding at most 25 bundles to 100 bundles. As shown in Fig. 7(a), when DTN nodes have more local storage, more bundles can be delivered to receivers. The average end-to-end delay also increases when the size of local storage increases. This is because with more local storage, some bundles that are dropped in the case of smaller storage could be buffered for longer time until they get a chance to be forwarded. Therefore, some of the delivered bundles may have been held in the network and experience longer queuing delays before reaching their destinations. Fig. 7(c) shows that for U-Multicast, ST-Multicast and DT-Multicast, the message delivery efficiency decreases when the buffer size increases, because more traffic is introduced into the networks as a result of the retransmissions of more buffered bundles. Still, OS-Multicast has the best message delivery ratio and the smallest delays but the worst efficiency.

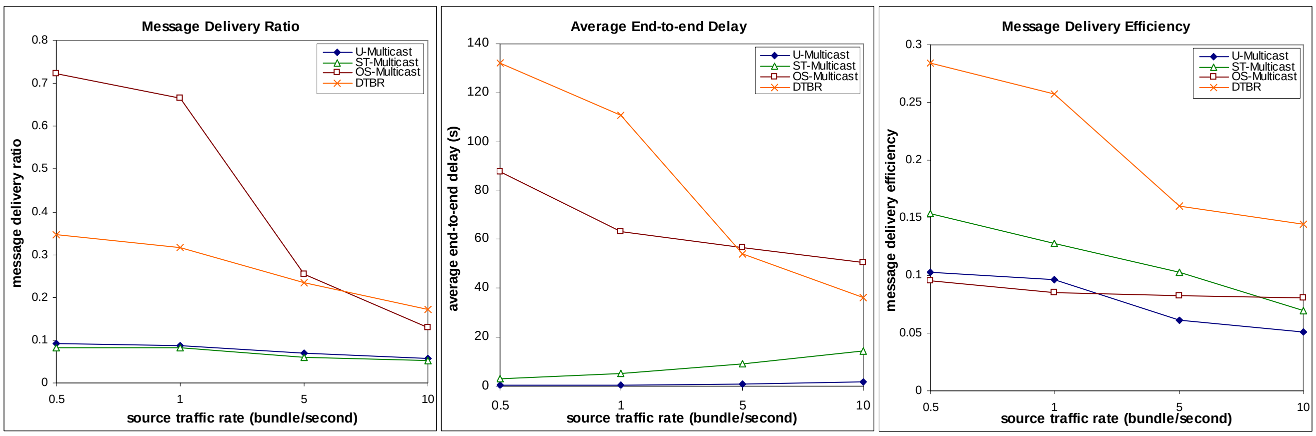
3) The impact of the percentage of DTN nodes

We are also interested in the impact of the scale of the DTN layer. As mentioned earlier, DTN functionalities are only implemented in those nodes that form the DTN layer. Our intuition in this experiment is that with more DTN nodes supporting multicast, the performance of all the multicast



(a) (b) (c)

Figure 9. Simulation results with the PSN trace



(a) (b) (c)

Figure 10. Simulation results of varying the source traffic rate with mobile nodes

strategies would become better.

We vary the percentage of the DTN nodes from 20% to 100% of the total nodes in the network. The results in Fig. 8 show that the message delivery ratios of all the approaches increase as more DTN nodes are deployed. However, their performances in terms of end-to-end delay and message delivery efficiency decrease. By studying the simulation traces, we find that when more DTN nodes are deployed, generally more potential paths to the receivers are discovered at each intermediate node. It indicates that there are larger possibilities for bundles to be forwarded to the destinations. Thus, it improves the message delivery ratios of different multicast approaches. However, with more DTN nodes, the average lengths of the discovered paths and the average local buffer usage of DTN nodes also increase. Therefore, bundles experience longer queuing delays. It explains why the average end-to-end delays become larger. Moreover, more retransmissions take place as more opportunistic paths are discovered. It incurs more redundancies and brings down the efficiencies of all the approaches.

4) Applying real-world DTN traces

Previously, we observed that different percentages of link up/down durations affect the performances of multicast strategies. In this simulation, we study the impact of different link up/down patterns. In our previous simulations, the probability of having a short inter-contact duration of a link is as same as that of having a long one (i.e., uniformly distributed). However, practical DTN experiments such as PSN (Pocket Switch Networks) [24] and DieselNet [25], have reported that the cumulative distribution function (CDF) of the inter-contact durations approximately follows a power-law distribution. It indicates that in practical cases DTN links are usually up for very short periods of time. Based on their observations, we retrieve the link up/down patterns from the trace files of PSN and apply them into our modified ns-2 simulator to make the inter-contact durations of DTN links follow the same power-law distribution.

Fig. 9 illustrates the results of using the PSN trace. In PSN, wireless devices called *iMotes* are put into the pockets of the conference attendees at IEEE Infocom 2005. A DTN link is set up only when two people encounter with each other in the same conference room. Fig. 9 shows that all multicast approaches perform much worse in terms of message delivery ratio and end-to-end delays than those shown in Fig. 6.

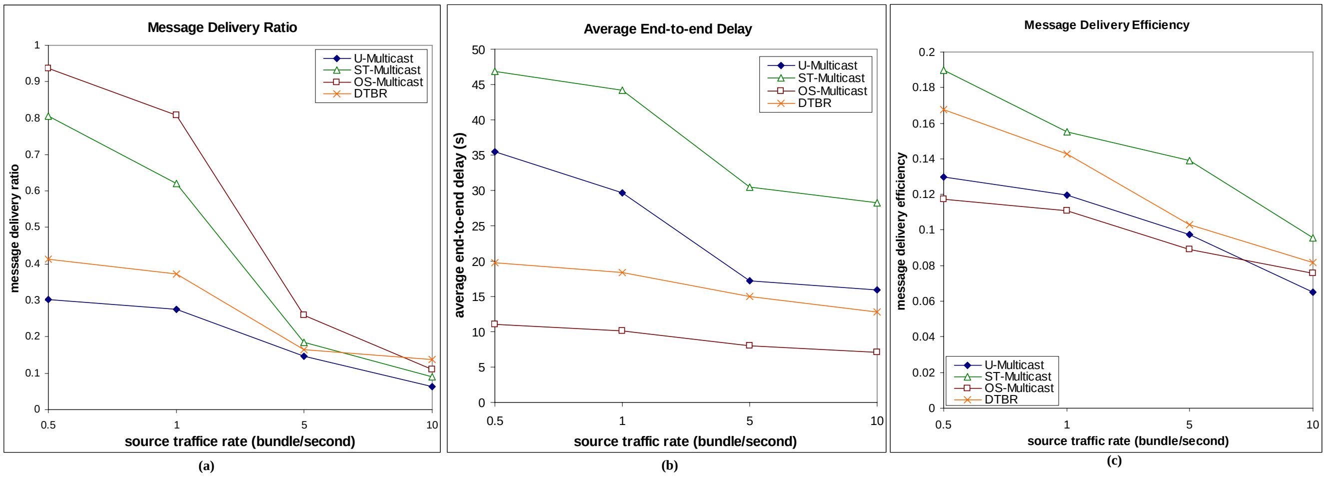


Figure 11. Simulation results of varying the source traffic rate with stationary nodes

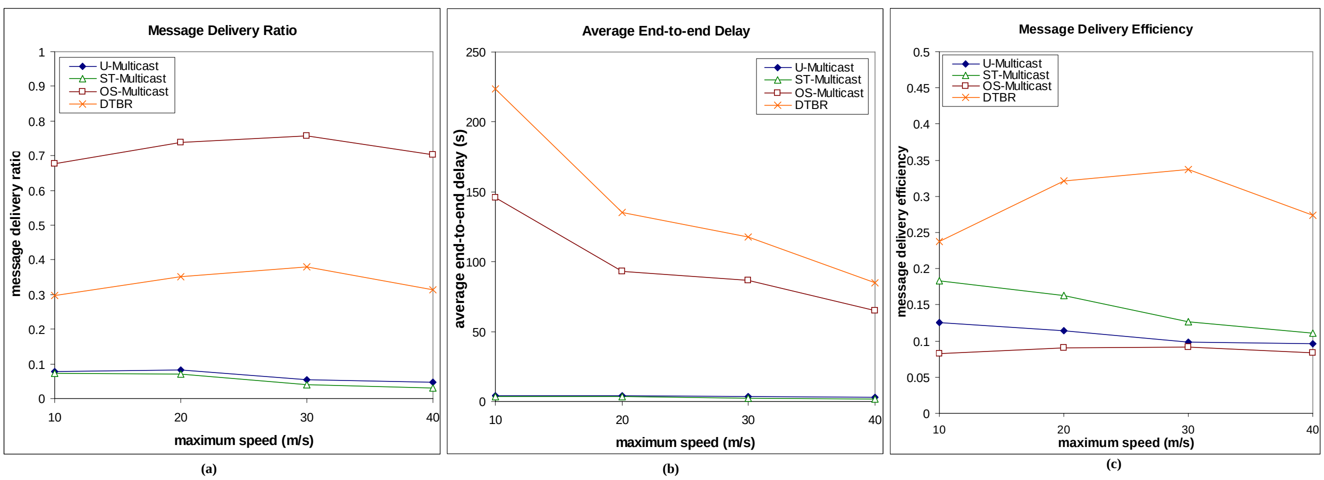


Figure 12. Simulation results of varying the maximum moving speed of nodes and low source traffic rate

Basically, fewer bundles are delivered to the destinations with larger delays by all multicast approaches. With the power-law distributed inter-contact durations, it becomes more difficult to deliver bundles when links are repeatedly down for a long period of time. In addition, bundles have to be buffered in the intermediate nodes for longer time before they can be successfully delivered to the receivers.

From all these simulations we conclude that: *i*) DT-Multicast strategy performs better than U-Multicast and ST-Multicast, especially when the networking links experience longer downtime durations frequently; and *ii*) Comparing to DTBR, OS-Multicast considers a possible trade-off between the message delivery ratio and efficiency. The next section will show that DTBR and OS-Multicast are appropriate for different DTN scenarios.

B. Results With Node Mobility & Under High Traffic Rate

In practice, DTN nodes like vehicles may move in the network areas rather than being stationary. In this simulation, we apply the mobility patterns retrieved from the ZebraNet trace [26], which records the location information of a group of zebras in Kenya and is regarded as another widely utilized mobility model for studying DTNs. Table II lists the simulation parameters. In this test, all the nodes are configured as DTN nodes and we vary the source traffic rate from 1 bundle per 2 seconds to 10 bundles per second, which is a relatively high traffic rate compared to that in the previous set of simulations. The simulation length is 3,000 seconds and the source only generates bundles during the first 1,000 seconds.

TABLE II. SIMULATION PARAMETERS

Parameter	Value
Number of nodes	25 (all are DTN nodes)
Area size	2000m × 2000m
Initial topology	uniformly randomly distributed
Number of multicast sources	1
Number of multicast receivers	4
Bundle size	512 bytes
Data rate	0.5 ~ 10 bundles/second
Bundle retransmission timer	5 seconds
MAC layer protocol	802.11
Transmission range	250m
Maximum movement speed	20 m/s

Fig. 10 shows the results. Not surprisingly, U-Multicast and ST-Multicast can only deliver very few bundles to the receivers as nodes move quickly. Neither can adapt well to the frequent topology variations caused by high node mobility. We observe that when the traffic rate at the source is high (> 5 bundles/second), DTBR outperforms OS-Multicast in all the metrics including the message delivery ratio, efficiency and average end-to-end delays. This is because in OS-Multicast each intermediate node will generate a large amount of redundant traffic to cover all the receivers under high traffic load. They compete with those freshly received bundles for the limited capacity of wireless links to be forwarded to the next hops in the multicast tree. Therefore, fresh bundles have few chances to be forwarded and suffer long queuing delays. This worsens the performance of OS-Multicast. However, when the traffic load is small, the redundancy introduced by OS-Multicast improves its bundle delivery performance because a bundle is considered as delivered when one of its copies reaches each destination. Based on our results, we recommend that system designers select DTBR when the source traffic rate is high to get the best multicast performance.

C. Other Results

We also investigate the performances of the DTN multicast strategies in the other two cases: stationary nodes with high source traffic rate and mobile nodes with low source traffic rate. The results are shown in Fig. 11 and Fig.12 respectively. Both use the same simulation setup as listed in Table II.

From Fig.11, we get the similar observations as that discussed in Section IV. B: the performance of OS-Multicast drops when the source traffic rate is relatively high. In Fig. 12, we fix the source traffic rate as 1 bundle/2 second but vary the maximum moving speed of DTN nodes from 10m/s to 40m/s. Fig. 12 shows that: *i*) the increase of node mobility can help DT-Multicast schemes deliver more bundles to the receivers faster, because it improves the chances of nodes to encounter a receiver and reduces the queuing delays; but *ii*) when the node mobility is high, it may worsen the message delivery ratio of DT-Multicast schemes because links in the discovered multicast structure can be more easily broken. How can take advantage of node mobility to assist multicast routing is out of the scope of this paper. The related discussions can be found in [41-43].

V. Background and Related Works

The characteristics of occasionally-connected networking in DTNs could be found in many networks that are subject to disruptions and disconnections. For example, the inter-planetary internet (IPN) [31] designed to support deep-space data transmissions is a typical scenario for DTNs in which wireless links connecting spaceships or space stations may be periodically unavailable. Data communications via LEO (Low Earth Orbit) satellites among several military bases across different continents is another example as data can be exchanged only when a satellite is visible to a military base. Wireless sensor networks which are deployed in hash environments may also suffer from frequent network partitions. Moreover, in the networks consisting of highly mobile vehicles without powerful antennas, the duration of an available link between two moving nodes will be very short [21].

A. DTN Unicast Routing Shemes

To overcome the frequent disruptions of end-to-end paths, DTN routing schemes are all conducted in a hope-by-hop store-and-forward manner. The basic idea is to take advantage of the local storage within each node and send bundles when appropriate forwarding opportunities come. However, how to be aware of, evaluate and utilize the data forwarding opportunities differs in different ways.

The existing DTN unicast routing methods can be divided into two categories: *knowledge-based* and *probability-based* routing schemes. Knowledge-based schemes assume that certain information about the networks such as prior knowledge of the link connectivity pattern, the geographic locations, or the node movement schedules have already been discovered and known by the networks. Therefore, routing decisions can be made using Dijkstra-like algorithms to decide when and how a message should be forwarded, by taking either the delay or the distance as the computing costs. ED (Earliest Delivery), MED (Minimum Expected Delay), EDLQ (Earliest Delivery with Local Queue) and EDAQ (Earliest Delivery with All Queues) methods proposed in [7], MV (Meetings and Visits) routing proposed in [14], and the message ferry routing and control schemes studied in [15, 16, 17] all belong to this category.

Probability-based DTN routing doesn't rely on any prior knowledge of the networks. In general, it requires that when each node encounters another it estimates the probability of successfully delivering a bundle to the destination by taking the other node as the next hop. A bundle is forwarded when the probability is better than a certain threshold. How to estimate that probability differs in different ways. FC (First Contact) in [7] is the simplest one; it always trusts the first available contact to any neighboring node and forwards the bundles. However, loops may be easily found in FC. Epidemic Routing [18] makes two nodes exchange all not previously seen messages once they encounter each other. The approach is based on the idea that randomly propagated message will eventually arrive at the destination. PROPHET (Probabilistic ROuting Protocol using History of Encounters and Transitivity) [19] extends the idea of Epidemic Routing by calculating the so-called delivery predictability based on the previous neighbors a node has encountered. A message is forwarded only when the next hop is predicted to have a better chance to directly encounter the destination in the future. However, such prediction could be inaccurate. Spray-and-Wait [20] is another extension of Epidemic Routing. It first sprays a message to a certain number of nodes, and then waiting for one of the copies to be directly forwarded to the destination. In this way, Spray-and-wait is able to reduce the large amount of overhead introduced by Epidemic Routing. Also, different message spraying strategies could be developed and utilized according to the network conditions. MaxProp [21] is one of the latest DTN unicasting methods. Its routing decision is made based on the path likelihood which estimates the delivery probability of choosing one path to forward a bundle. The path likelihood varies according to the changes of DTN environments. Compared to those knowledge-based routing approaches, the probability-based DTN routing methods often have non-global-optimal route selections because routing is always decided based on local information.

B. Multicast Routing Schemes

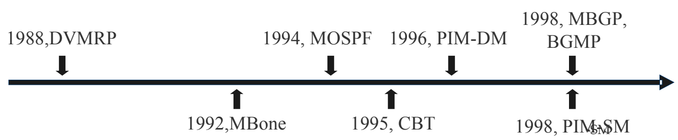


Figure 13. The time table of Multicast routing schemes in the Internet

The idea of IP multicast was first introduced by Steve Deering in his dissertation in 1988 [33]. In the first generation of Multicast Backbone (MBone) in 1992 [34], DVMRP [9] was applied to build up the multicast tree in a broadcast-and-prune manner. After that, many multicast protocols have been proposed for the Internet, which can be divided into two categories: source-rooted tree schemes (e.g., MOSPF [8] and PIM-DM [35]) and shared tree schemes (e.g. CBT [36] and PIM-SM [37]). The major difference between these two is whether to maintain the multicast tree at the source or at the rendezvous point (RP) which is shared by multiple trees in order to save more management and network resources.

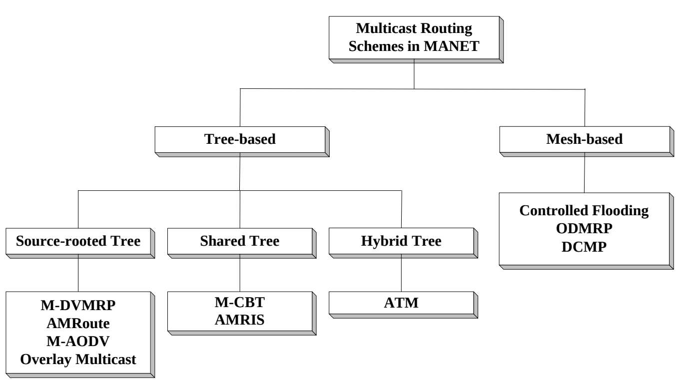


Figure 14. Multicast routing schemes in MANET

In MANET, the multicast routing schemes can be divided into two categories based on the choice of maintaining a multicast mesh or a tree. The advantage of the mesh-based schemes such as ODMRP (On-Demand Multicast Routing Protocol) [38] is the high message delivery ratio achieved by forwarding data via multiple paths in the mesh, while that of the tree-based schemes such as AMRoute (Ad hoc Multicast Routing Protocol) [39] is the high message delivery efficiency. Moreover, ATM [40] is a hybrid tree-based approach that can switch between the source-rooted tree and shared tree by comparing the distances from the receivers to the source to that to the RP.

To our best knowledge, DTBR and OS-Multicast are the two existing multicast routing schemes proposed for DTNs. However, the controlled flooding schemes in [44] can be extended to conduct the one-to-many data communications in sparse mobile wireless networks. Also, ideas in [44] can be applied to OS-Multicast to improve its efficiency.

VI. CONCLUSIONS

In this article, we have discussed several multicast strategies including U-Multicast, ST-Multicast, and DT-Multicast that are applicable to disruption tolerant networks. We focus on studying two DT-Multicast routing schemes: DTBR and OS-Multicast, which are able to dynamically adjust the multicast structure in a hop-by-hop manner according to the current network conditions. Performance comparisons among these multicast methods are then done by simulations. Our results show that: *i*) DT-Multicast approaches significantly outperform ST-Multicast and U-Multicast for DTNs, especially when networking links are only up for very short periods of time and nodes may move fast. In most cases, DT-Multicast can achieve the best message delivery ratio with the smallest delays. *ii*) The performances of DTN multicast routing schemes are sensitive to some network configuration parameters, such as the number of DTN nodes within the networks and the local storage size inside each DTN node. In general, we recommend a system designer deploy more DTN nodes with larger buffer to achieve better message delivery ratio. However, there is a trade-off between the performance enhancements and the costs of implementation, deployment, and resources. *iii*) DTBR and OS-Multicast have different advantages. The fundamental design purpose of OS-Multicast is to improve the multicast reliability and the utilization of opportunistic links by introducing more redundancy. Thus, OS-Multicast is more appropriate when the traffic load is small. When the traffic load becomes high, DTBR performs better in terms of the message delivery efficiency.

ACKNOWLEDGMENT

This work has been supported by DARPA under Contract W15P7T-05-C-P413. 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] M. Chuah, L. Cheng, and B.D. Davison, “Enhanced disruption and fault tolerant network architecture for bundle delivery (EDIFY)”, *Proceedings of IEEE Globecom*, pp. 807-812, November 2005.

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

[3] DARPA Disruption Tolerant Networks program http://www.darpa.mil/ato/solicit/dtn/, accessed on Nov. 3rd, 2006.

[4] S. Jain, M. Demmer, R. Patra, K. Fall, “Using Redundancy to Cope with Failures in a Delay Tolerant Network”, *Proceedings of SIGCOMM’05*, Augest 2005.

[5] R. Malladi and D.P. Agrawal, “Current and future applications of mobile and wireless networks”, *Communications of the ACM*, Vol. 45, pp. 144-146, 2002.

[6] 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.

[7] S. Jain, K. Fall, R. Patra, "Routing in a delay tolerant networking", *Proceedings of SIGCOMM’04*, Augest 2004.

[8] J. Moy, “Multicast extensions to OSPF”, IETF RFC 1584, 1994.

[9] D. Waitzman, C. Partridge and S. Deering, “Distance vector multicast routing protocol (DVMRP)”, *IETF RFC* 1075, 1988.

[10] J. Xie, R.R. Talpade, A. Mcauley, and M.Y. Liu, “AMRoute: ad hoc multicast routing protocol”, *Mobile Networks and Applications*, Vol. 7, Issue 6, pp. 429-439, 2002.

[11] S.H. Bae, S.-J. Lee, W. Su, and M. Gerla, “The design, implementation, and performance evaluation of the on-demand multicast routing protocol in multihop wireless networks”, *IEEE Network*, pp.70–77, January 2000.

[12] W. Zhao, M. Ammar, and E. Zegura, “Multicasting in delay tolerant networks: semantic models and routing algorithms,” *Proceeding of Sigcomm’05 Workshop in DTN*, August 2005.

[13] Q. Ye, L. Cheng, M. Chuah, and B. Davison, “OS-multicast: On-demand Situation-aware Multicasting in Disruption Tolerant Networks”, *Proceedings of IEEE 63rd VTC*, Vol. 1, pp. 96-100, Melbourne, Australia, May 2006.

[14] B. Burns, O. Brock, and B.N. Levine, “MV routing and capacity building in disruption tolerant networks”, *Proceedings of IEEE INFOCOM*, pp. 398–408, March 2005.

[15] W. Zhao and M. Ammar, “Message Ferrying: Proactive Routing In Highly-Partitioned Wireless Ad Hoc Networks”, *Proceedings of IEEE Workshop on Future Trends in Distributed Computing Systems*, May 2003.

[16] W. Zhao, M. Ammar, and E. Zegura, “A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad hoc Networks”, *Proceedings of ACM Mobihoc*, May 2004.

[17] W. Zhao, M. Ammar, and E. Zegura, “Controlling the mobility of multiple data transport ferries in a delay-tolerant network”, *Proceedings of IEEE INFOCOM’05*, March 2005.

[18] A. Vahdat and D. Becker, “Epidemic Routing for Partially-Connected Ad Hoc Networks”, *Duke Technical Report CS-2000-06*, July 2000.

[19] A. Lindgreny, A. Doria, and O. Schelén, “Probabilistic Routing in Intermittently Connected Networks”, *Proceedings of the First International Workshop on Service Assurance with Partial and Intermittent Resources (SAPIR 2004)*, Augest 2004.

[20] T. Spyropoulos, K. Psounis, and C.S. Raghavendra, “Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks”, *Proceeding of Sigcomm’05 Workshop in DTN*, August 2005.

[21] J. Burgess, B. Gallagher, D. Jensen, B.N. Levine, “MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networking”, *Proceedings of IEEE INFOCOM’06*, April 2006

[22] V. Cerf, "Delay Tolerant Network Architecture", draft-irtf-dtnrg-arch-04.txt, December 2005.

[23] Network Simulator 2, ns2. http://www.isi.edu/nsnam/ns/

[24] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, “Pocket Switched Networks and Human Mobility in Conference Environments”, *Proceeding of Sigcomm Workshop in DTN*, August 2005.

[25] B. Burns, O. Brock, and B. Neil Levine, “Autonomous Enhancement of Disruption Tolerant Networks”, *Proceedings of IEEE International Conference on Robotics and Automation*, May 2006.

[26] Y. Wang, S. Jain, M. Martonosi, and K. Fall, “Erasure Coding Based Routing for Opportunistic Networks”, *Proceeding of Sigcomm Workshop in DTN*, August 2005.

[27] R. Bhatia, L. Li, “Characterizing Achievable Multicast Rates in Multi-Hop Wireless Networks ”, *Proceedings of the Sixth ACM International Symposium on Mobile Ad Hoc Networks (MobiHoc’05)*, May, 2005.

[28] S. Farrell, V. Cahill. "Delay and Disruption Tolerant Networking," ISBN 1-59693-063-2, Artech House, 2006.

[29] S. Farrell, V. Cahill, D. Geraghty, I. Humphreys, and P. McDonald, "When TCP Breaks: Delay- and Disruption-Tolerant Networking." *IEEE Internet Computing*, vol. 10, no. 4, 2006, pp. 72-78.

[30] Z. Zhang, "Routing in Intermittently Connected Mobile Ad Hoc Networks and Delay Tolerant Networks: Overview and Challenges." *IEEE Communications Surveys and Tutorials*. Jan, 2006.

[31] S. Burleigh, “Delay-Tolerant Networking -- An Approach to Interplanetary Internet”, *IEEE Communications Magazine*, June 2003.

[32] C. Gui and P. Mohapatra, “Overlay Multicast for MANETs using dynamic virtual mesh”, *Wireless Networks*, vol. 13, pp. 77-91, 2007.

[33] Stephen Deering and D. Cheriton, “Multicast Routng in Datagram Internetworks and Extended LANs”, *ACM Transcations on Computer Systems*, pp. 85-111, May 1990.

[34] Stephen Casner and Stephen Deering, “First IETF Internet Audiocast”, *ACM Computation Communication Review*, pp. 92-97, July 1992.

[35] Stephen Deering, D.Estrin, D. Farinacci, V. Jacobson, G. Liu and L. Wei, “PIM Architecture for Wide-aree Multicast Routing”, *IEEE/ACM Transcation on Networking*, pp. 153-162, April, 1996.

[36] Tony Ballardie, Paul Francist and Jon Crowcroft, “Core Based Tree: An Architecture for Scalable Inter-domain Multicast Routing”, *preceeding of ACM SIGCOM'95*, pp. 85-95, 1995.

[37] D. Estrin, “Protocol Independent Multicast Sparce Mode (PIM-SM): Protocol Specification”, *IETF RFC 2362*, 1998.

[38] S. H. Bae, S. Lee, W. Su, and M. Gerla, “The Design, Implementation, and Performance Evaluation of the on-Demand Multicast Routing Protocol in Multihop Wireless Networks”, *IEEE Network*, pp. 70–77, January 2000.

[39] J. Xie, R. Talpade, A. Mcauley, and M. Liu, " AMRoute: ad hoc multicast routing protocol", *Mobile Networks and Applications*, Vol. 7, Issue 6, pp. 429-439, 2002.

[40] M. Gerla, C. Chiang and L. Zhang, “Tree multicast strategies in mobile, multihop wireless networks”, *Mobile Networks and Applications*, Vol.4, pp.193–207, 1999.

[41] S.-J. Lee, W. Su, and M. Gerla, “Wireless Ad hoc Multicast Routing with Mobility Prediction”, *ACM/Kluwer Mobile Networks and Applications*, 2001.

[42] J. Kong, X. Hong, M. Y. Sanadidi, and M. Gerla, “Mobility Changes Anonymity: Mobile Ad Hoc Networks Need Efficient Anonymous Routing”, *Proceeding of The Tenth IEEE Symposium on Computers and Communications (ISCC)*, La Manga del Mar Menor, Cartagena, SPAIN, June 27-30, 2005.

[43] X. Zhang, G. Neglia and J. Kurose, “Performance Modeling of Epidemic Routing”, *Proceeding of IFIP Networking 2006*, May, 2006.

[44] K. A. Harras, K. C. Almeroth and E. M. Belding-Royer, “Delay Tolerant Mobile Networks (DTMNs): Controlled Flooding in Sparse Mobile Networks”, *Proceedings of ACM SIGCOMM Workshop on Challenged Networks (CHANTS)*, September, 2006.