Context-Aware Multicast Routing Scheme for Disruption Tolerant Networks

Peng Yang
Lehigh University
Bethlehem, PA 18015
610-758-4061
pey204@lehigh.edu

Mooi Choo Chuah
Lehigh University
Bethlehem, PA 18015
610-758-4061
chuah@cse.lehigh.edu

ABSTRACT
Disruption Tolerant Networks (DTNs) are emerging solutions to networks that experience frequent network partitions and large end-to-end delays. Several schemes have been proposed for multicast routing in DTNs assuming the availability of different amount of knowledge about network topology etc. In this paper, we propose a node-density based adaptive multicast routing scheme which can handle different network scenarios than the existing multicast delivery schemes for DTNs that we are aware of. Our scheme can address the challenges of opportunistic link connectivity in DTNs. Simulation results show that our CAMR scheme performs better than the DTBR and OS-multicast schemes. The CAMR scheme can achieve a better message delivery ratio, with higher transmission efficiency and similar delay performance especially when the nodes are very sparsely connected.

Categories and Subject Descriptors
C.2.2 [**Network Protocols**]: Routing Protocols

General Terms
Algorithms, Performance, Design.

Keywords
Routing, Delay Tolerant Network, Multicast, Adaptive

1. INTRODUCTION
There are emerging network scenarios where an instantaneous end-to-end path between a source and destination may not exist because links between nodes may be opportunistic. New architecture and communication protocols need to be designed to allow nodes in such challenging network environments to communicate with one another. There is ongoing research [2, 3, 4, 5] on disruption tolerant networks (DTNs) that addresses such challenges. DTNs have a broad range of potential applications e.g. military battlefields [7], vehicular communications [15,16], deep-space communications [8], habitat monitoring [9], and Internet access in rural areas [10].

Many DTN applications need multicast service. For example, in military battlefields, it is essential that orders from a command center can be quickly and reliably disseminated to a group of field commanders. It is also helpful to let different groups of soldiers share information of their surrounding environments. However, traditional multicast methods proposed for the Internet (e.g., MOSPF [11] and DVMRP [12]) or mobile ad hoc networks (e.g., AMRoute [13] and ODMRP [14]) are not suitable for DTNs, due to the frequent network partitions and sparse connectivities. With partitions and poor connectivity among nodes, it is difficult to maintain a source-rooted multicast tree during the lifetime of a multicast session. The application data suffer from large end-to-end delivery latencies due to the disruptions caused by the frequent link breakages. In fact, the traditional approaches may fail to deliver a message when the link is highly unavailable (e.g. ~80%). In [2], the authors investigate several multicast routing algorithms e.g. Unicast Multicast (U-Multicast), static tree-based routing (STBR), dynamic tree-based routing (DTBR), a unicast-based routing, and group-based routing. In [17], the authors propose an on-demand situation-aware multicast (OS-multicast) approach which is a dynamic tree-based method that integrates DTN multicasting with the situation discovery mechanism provided by the underlying network layer. Their simulation results show that OS-multicast can achieve smaller delays and better message delivery ratios than other schemes e.g. DTBR. OS-multicast also achieves higher efficiency when the probability of link unavailability is high and the duration of link downtime is large. However, these approaches rely on the opportunistic connectivities among nodes for delivery without making use of additional information like node location and node velocities. We are interested in understanding the performance improvement one can get from a multicast routing scheme that makes use of such information. We also want to design a scheme that is adaptive to the network environments that are dynamically changing e.g. densely connected nodes in the battlefield may become very sparsely connected when they face obstacles while they move around. Thus, in this paper, we design a node-density based adaptive multicast routing scheme for DTNs that uses information like node location and velocities. We compare our CAMR scheme with U-Multicast, DTBR, and OS-Multicast via simulation studies. Our results indicate that CAMR is more flexible than the other schemes. Our CAMR scheme achieves the highest delivery ratio and data efficiency.

The rest of the paper is organized as follows. In Section 2 we present descriptions of the network model and multicast model of DTNs. In Section 3, we briefly describe DTBR and OS-Multicast schemes; we describe our CAMR scheme. Performance evaluations are presented in Section 4. Section 5 summarizes our contributions.

2. SYSTEM MODELS

2.1 Network Model

A DTN is an overlay network that is built upon underlying networks e.g. wireless ad hoc networks. Its network architecture is based on the asynchronous message (called bundle) forwarding paradigm presented in [1]. Only those nodes that implement the DTN functionalities e.g. sending and receiving bundles are considered DTN nodes, while the others are denoted as normal nodes. A DTN link may span several underlying links. Fig. 1 depicts a simple DTN example.

In the DTN layer, bundles are transmitted in a store-and-forward manner hop by hop. Each DTN node has finite-size buffers. To ensure reliable delivery, a custodian transfer feature has been proposed [4]. More details about our DTN architecture can be found in [5].

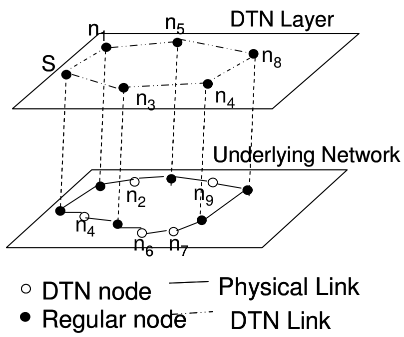

Figure 1. A simple DTN example

2.2 Multicasting Model

Multicast in DTNs is defined as the one-to-many or many-to-many bundle transmissions among a group of DTN nodes. A multicast source uses a group name or an explicit list of the names of individual DTN multicast receivers as the destination address for delivering multicast bundles. The later approach may not be scalable when the number of DTN multicast receivers grow large. To achieve scalability, the intermediate nodes may have to cache information about multicast receivers similar to the Internet multicast approach.

3. MULTICAST ROUTING APPROACHES

3.1 Existing Multicast Approaches for DTN

As we mentioned before, DTNs suffer from frequent network partitions. The dynamically changing network topology makes the maintenance of any multicast tree in DTNs challenging. The performance of different multicast approaches relies on the knowledge of network conditions that DTN nodes can discover and disseminate among one another. Therefore, situational information needs to be collected and disseminated continuously so that new information can be used to adjust the message delivery paths. Different policies that utilize different amount of situational information for routing decisions can be created for different applications with varying quality of service requirements.

An obvious multicast delivery approach is to send a separate bundle to each of the receivers. This is referred to as U-Multicast[2]. In addition, there are two existing approaches for supporting multicast communications in a DTN. A simple example of these two approaches is illustrated in Fig.2.

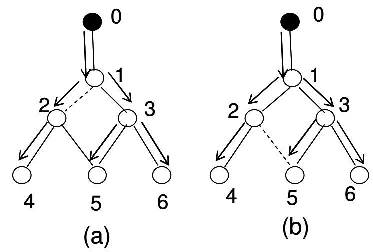

Figure 2. Multicast approaches in DTN (a) DTBR, (b) OS-multicast: when link 2→5 is unavailable and link 3→5 becomes available, node 3 will take advantage of the current available link immediately.

- **DTBR** [2]: This is a dynamic tree-based multicasting algorithm designed for DTNs. In DTBR, the upstream node will assign the receiver list for its downstream neighbors based on its local view of the network conditions. The downstream nodes are required to forward bundles only to the receivers in the list, even if a new path to another receiver (not in the list) is discovered. For example, in Figure 2(a), say link 1-2 is unavailable when the multicast bundle reaches node 1. Then, node 1 will use node 3 to deliver to nodes 5 and 6 and store a copy of the bundle so that node 1 can send to node 2 when the link 1-2 becomes available again since this is the only route (via link 1-2) that node 1 knows of to reach node 4. DTBR assumes that each node has complete knowledge or the summary of the link states in the network. However, this is hard to achieve in practical scenarios.
- **On-demand Situation-aware multicast** (OS-multicast) [17] Like the DTBR scheme, OS-multicast is also a dynamic tree-based multicast approach. A unique multicast tree is constructed for each bundle and the tree is adjusted at each intermediate DTN node according to the current network conditions. When a DTN node receives a bundle, it will dynamically adjust an initially constructed tree based on its current knowledge of the network conditions. Via such adjustments, any newly discovered path will be quickly utilized. For example, in Figure 2(b), the link between 2-5 is broken but when the bundle reaches node 3, node 3 knows that it can reach node 5 and 6. So, it will send a copy to both nodes 5 and 6. The downside of the OS-multicast approach is that a receiver may receive multiple copies of the same bundle.

3.2 Context Aware Multicast Routing (CAMR) Scheme

In this section, we describe the CAMR scheme we proposed. This scheme consists of five components: (a) Local Node Density Estimation, (b) 2-Hop Neighbor Contact Probability Estimate, (c) Route Discovery, (d) Route Repair, and (e) Data Delivery. The pseudo codes for the CAMR scheme are shown in Figure 3.

a) Local Node Density Estimation

We assume that every node periodically (say every 20 seconds) broadcasts a neighbor discovery message using regular power transmission (say with a transmission range of 250 m). A node that hears a neighbor discovery message will respond with a unicast neighbor response message after some random backoff delay (to prevent broadcast storm). The neighbor response message includes this node's information (e.g. identifier, location, and velocity) and this node's 1-hop neighbor's information (e.g. the neighbor's identifier, contact probability, location, and velocity). Thus, each node can estimate the average number of neighbors it has, denoted as n_d . If n_d drops below a threshold *K*, then the node sets a sparsely connected flag.

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

Upon expiration of BROADCAST_TIMER do
   localDensityDetection();

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

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

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

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

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

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

Figure 3 (a): Pseudo Codes for Local Node Density Estimate

b) 2-Hop Neighbor Contact Probability Estimate

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

c) Route Discovery

A source initiates a route discovery process if it cannot reach any of the receivers to which the multicast traffic needs to be delivered. Before any intermediate node re-broadcasts a route request message it receives, it first checks to see if its sparsely connected flag is set. If the flag is set, then the node rebroadcasts the route request message using high power transmission (e.g. at a level that results in a transmission range of 375m or 500m). Otherwise, the intermediate node re-broadcasts the route request using regular power. Any node that receives a high power route request will make a note since that node needs to issue a high power route reply when it hears a response back from a downstream node.

Assume that S is the multicast source and there are eight multicast receivers as shown in Figure 4. S knows it can reach R_6 using its 2-hop neighborhood information so S does not need to issue any route request for R_6. The route request issued by S for the other seven receivers is flooded by intermediate nodes using regular power until it reaches node R_6 and n_3. R_6 and n_3 will re-broadcast the route request using high power since their observed local node densities drop below the threshold *K*. When the route request eventually reaches any intended multicast receiver, it will issue a route reply. Through this process, S can eventually construct a merged multicast delivery tree after hearing route replies from downstream nodes for all the eight receivers. Note that this multicast tree is not static as in wired multicast or static wireless mesh network scenario. Note further that the location and velocity information of the nodes sending route request and route reply messages are piggybacked. For example, when R_6 receives a route reply from n_4, R_6 will record the location and velocity information of n_4 and that n_4 is the next hop node for delivering bundles from this multicast session. Later, when S sends multicast bundles, S will piggyback a multicast receiver list in all multicast bundles that are sent out. Thus, R_6 knows that it is responsible for delivering the multicast messages to R_1, R_5, R_7 and R_8. Upon receiving the multicast bundles, R_6 travels closer to n_4 so that the multicast bundles can be delivered using regular power transmission. Since the location and velocity information is included in the route reply from n_4 to R_6, R_6 can estimate where it can reach n_4 with regular power transmission. Note that R_6 may decide to travel towards n_4 only after receiving multiple multicast bundles (aka batch delivery). Similar actions are taken by n_3 to deliver multicast bundles to n_6.

There are several optimizations that one can make e.g. the source can flood a multicast route request where the identifiers of all receivers are included rather than sending individual route request, the intermediate nodes can merge the route replies from different downstream nodes that can reach different multicast receivers. In this paper, our simulation study does not include these optimizations.

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

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

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

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

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

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

PROCEDURE composeRouteDiscovery()
    Create Route_Discovery packet RDp;
```

Figure 3(b): Pseudo codes for Route Discovery

d) Route Repair

Custodian transfer feature [19] is turned on in our CAMR scheme. This means that after a node n_i receives a custodian acknowledgment from a downstream node, n_j, n_i can remove the acknowledged bundle from its storage.

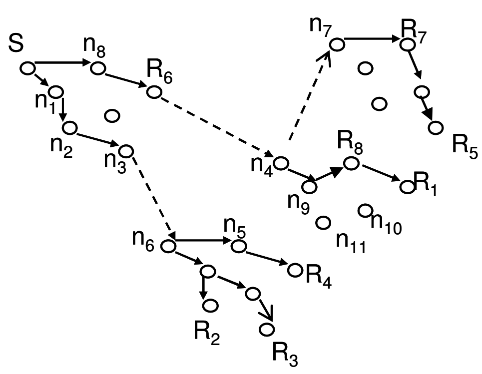

Figure 4. CAMR scheme

With this custody transfer feature enabled, one can use local route repairs when the multicast tree is broken as a result of node mobility. Let us assume that when multicast bundles arrive at node n_4 in Figure 4, n_9 moves away. There are two ways whereby n_4 can repair the route: (a) n_4 can issue route request to find a route to R_8, or (b) n_4 can make use of the location and velocity information to travel closer to n_9 and not incur extra route discovery messages for local repair. We refer to the version where n_4 issues a route request message to perform the local route repair as CAMR-I and the version where n_4 uses the location and velocity information to perform route repair as CAMR-II. The CAMR-II scheme will incur smaller routing overhead than the CAMR-I scheme.

```
Route Repair
Upon loss of acknowledgment from the next_hop along the data route do
    routeRepair(next_hop);

PROCEDURE routeRepair(next_hop)
    CAMR-I:
            RDp ← composeRouteDiscovery();
            if LOCAL_DENSITY==0 then
              broadcast RDp with high transmission power;
            else
              broadcast RDp with regular transmission power;

    CAMR-II:
            if lookUpLocation(next_hop) == TRUE then
              move close to the estimated next_hop.position;
            else
              goto CAMR-I;

PROCEDURE lookUpLocation(next_hop)
    next_hop.position ← check old location and velocity of next_hop in neighbor table
                        and estimate current location of next_hop
    distance ← ||next_hop.position-my.position||;
    if distance > MOVING_THRESHOLD          //if distance is too long, give up and
      return FALSE;                         //use CAMR-I scheme
    else
      return TRUE;
```

Figure 3(c): Pseudo codes for Route Repair

e) Data Delivery

For data delivery, an extra header is piggybacked to each data bundle. The header contains information on the identifies of the receivers to which a particular multicast bundle needs to be delivered. Any intermediate node that supports CAMR scheme will duplicate the bundle if the node discovers that it is the branching point. A node acts as a message ferry during data delivery if it uses a high power route request to reach a downstream node.

4. PERFORMANCE EVALUATION

To evaluate the performance of different multicast algorithms, we implemented U-multicast, OS-multicast, DTBR, and CAMR (both versions I & II) in the ns2 simulator. The performance metrics that are used to compare different multicast routing approaches are:

i) *message delivery ratio*, which is defined as the number of unique multicast bundles which successfully arrive at all the receivers over the total number of bundles which are expected to be received;

ii) *data efficiency*, which is the ratio between the unique bundles received by the receivers and the total data traffic generated in the networks;

iii) *overall efficiency*, which is the ratio between the unique bundles received by the receivers and the total traffic generated (both data and control packets) in the networks; and

iv) *average message delay*, which is the average of the end-to-end bundle delivery latencies for each algorithm (we observe similar delay performance results when we use the metric of median delay).

Note that for the overall efficiency computation, we assume that the power required to transmit a data packet is a linear function of the packet size. Each route request or reply message that is transmitted at high power is counted as (k=(r2/r1)^4) times that of a route request/reply message that is transmitted at regular power where r2 and r1 is the transmission range of the high and regular power transmission respectively e.g. k=16 when the high power transmission range is 500 m but the regular power transmission range is 250m. The neighbor discovery interval is set at 20 seconds, and β is set at 0.8 in all our experiments. In our experiments, unless otherwise stated, we use a network with 40 nodes deployed randomly in a geographical area that ranges from 1000x1000 m^2 to 4000x4000 m^2. All nodes are DTN nodes. DSR is used as the routing approach for the underlying ad hoc networks for DTBR and OS-Multicast schemes. And situational awareness is achieved through the communication between the DTN multicasting agent and the DSR routing agent. The MAC layer is IEEE 802.11 with radio transmission range that varies from 250 m to 500 m. In our first set of experiments, we use random waypoint mobility model. In our second set of experiments, we use a scaled ZebraNet [18] mobility model. The speed of the nodes not acting as message ferry is chosen uniformly between 1 m/s and 5 m/s. When a node acts as a message ferry, it moves with a speed of 15 m/s.

In our first set of experiments, we evaluate the impact of high power transmission range on the performance of the CAMR scheme. The regular transmission range is set at 250 m and the high power transmission range is set to either 500 m or 375 m. Each multicast source sends messages at a rate of 0.25 msg/sec. In this first set of experiments, we use one multicast source with 12 receivers. In our second set of experiments, we evaluate the impact of mobility model on the performance of the CAMR scheme. In the third set of experiments, we evaluate the impact of multicast group size. In our fourth set of experiments, we evaluate the impact of traffic load on the performance of CAMR scheme. In our fifth set of experiments, we compare our scheme with U-Multicast, DTBR and OS-Multicast schemes.

4.1 Impact of High Power Transmission Range

Tables 1 & 2 tabulate the results we obtain for the 1st set of experiments with the high power transmission range set at 500 m using CAMR-I or CAMR-II. From the results in Tables 2 &3, one can observe that CAMR-II scheme achieves slightly higher delivery ratio and lower delivery latencies compared to CAMR-I scheme. Hence for the rest of our simulation studies, we only use CAMR-II scheme and will refer to it as the CAMR scheme.

Figures 5, 6 and 7 plot the delivery ratio, average (and 95%) delay, and the data efficiency (and overall efficiency respectively that we obtained using CAMR-II when different high power transmission range is used. With the high power transmission range set to 500 m, the CAMR scheme can achieve a delivery ratio of 81.7% to 92.3% as the node density varies from 10^{-6} (the 4000x4000 m^2 scenario) to 2.5x 10^{-5} (the 1000x1000 m^2 scenario). The data efficiency decreases from 0.4 to 0.29 as the node density decreases. As the network becomes sparser, the 95% message delivery time increases by almost 7 times (it varies from 1332 seconds to 8907 seconds). With a high power transmission range of 375m, the data efficiency and the delivery ratio drops a little bit but the overall efficiency is improved. The message delivery latency increases by 7-29% when the node density decreases from 10^{-5} to 10^{-6}.

We also have results using different values for the neighbor discovery interval and β for the 4000x4000 m^2 scenario. Setting the neighbor discovery interval shorter improves the delivery ratio slightly but decreases the data and overall efficiencies due to higher transmission overhead e.g. with an interval of 10 seconds, we can get 86.2% delivery ratio versus 81.7% using an interval of 20 seconds. Similarly, a smaller β allows old information to be forgotten faster and hence improves the delivery ratio slightly (e.g. with β=0.4, we can get a delivery ratio of 85.6% versus 81.7% with β = 0.8).

Table 1: Performance of CAMR-I scheme (500m)

Simulation Area	Delivery Ratio	Average Delay	95% Message Delay(sec)	Data Efficiency	Overall Efficiency
1000x1000	91.5%	234	1135	0.52	0.19
2000x2000	90.2%	1607	5131	0.37	0.21
3000x3000	85.9%	3082	7500	0.12	0.06
4000x4000	80.4%	4332	10000	0.30	0.06

Table 2: Performance of CAMR-II scheme (500m)

Simulation Area	Delivery Ratio	Average Delay	95% Message Delay(sec)	Data Efficiency	Overall Efficiency
1000x1000	92.3%	196	1332	0.40	0.29
2000x2000	91.2%	1097	5446	0.35	0.22
3000x3000	86.4%	1749	7232	0.22	0.08
4000x4000	81.7%	3229	8907	0.29	0.08

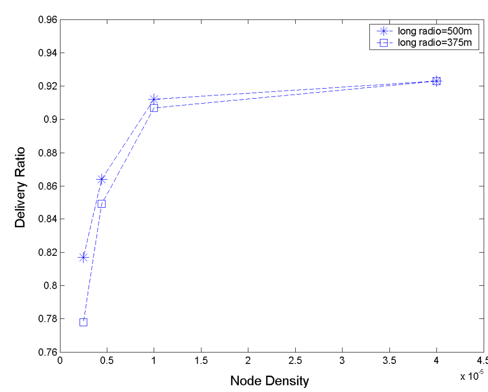

Figure 5: Delivery Ratio of CAMR-II scheme

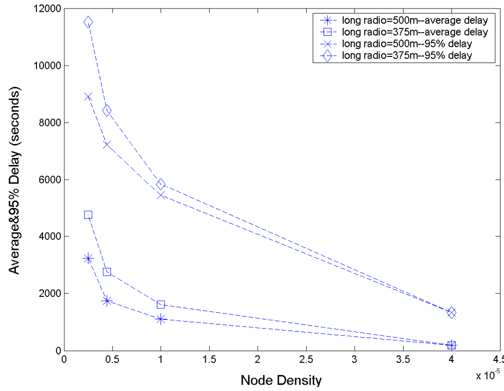

Figure 6: Average/95% Delay of CAMR-II scheme

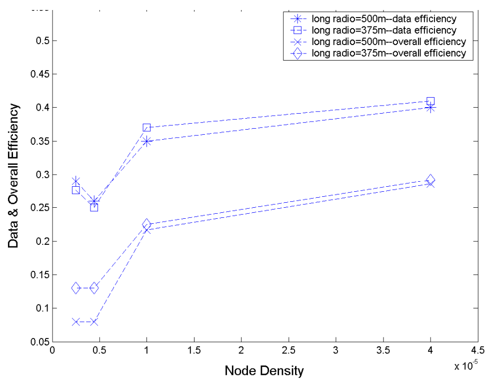

Figure 7: Data/Overall Efficiency of CAMR-II scheme

4.2 Impact of Mobility Model

In the first set of experiments, we use random waypoint mobility model. We repeated similar experiments using scaled Zebranet mobility model. Our results are tabulated in Tables 3 & 4.

Table 3: Performance of CAMR with ZebraNet Mobility (with high power transmission range = 500 m)

Simulation area	Delivery ratio	Average delay	95% message delay(sec)	Data efficiency	Overall efficiency
1000x1000	92.3%	46	183	0.27	0.16
2000x2000	92.5%	558	1063	0.26	0.18
3000x3000	88.9%	1515	6121	0.40	0.19
4000x4000	82.4%	4107	12562	0.44	0.14

Table 4: Performance of CAMR with ZebraNet Mobility (with high power transmission range= 375 m)

Simulation area	Delivery ratio	Average delay	95% message delay(sec)	Data efficiency	Overall efficiency
1000x1000	92.3%	45.6	183	0.27	0.16
2000x2000	92.5%	558	1063	0.26	0.18
3000x3000	88.3%	1597	6245	0.39	0.23
4000x4000	81.5%	4195	13410	0.5	0.23

From Tables 2 & 3, we see that the CAMR scheme achieves slightly higher delivery ratio with ZebraNet mobility model than with random waypoint mobility model when the network becomes sparser. In addition, the data efficiency and overall efficiency is higher with ZebraNet mobility scenario than with the random waypoint mobility model. The same conclusion applies when we compare the two sets of results with a high power transmission range of 375 m.

4.3 Impact of Group Size

Next, we investigate how the size of the multicast group affects the performance. We repeat the experiment in subsection 4.1 (with random waypoint mobility model and an area of 4000x4000 m^2) using different group sizes. Our results are plotted in Figures 8,9, and 10.

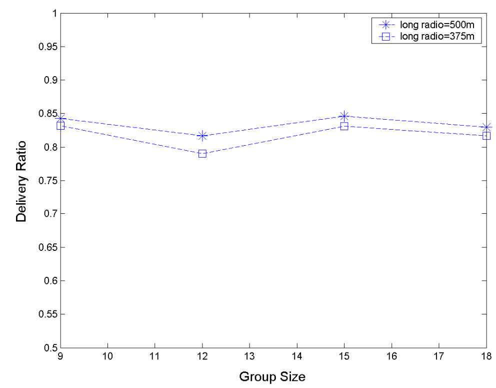

Figure 8: Delivery Ratio with different group sizes

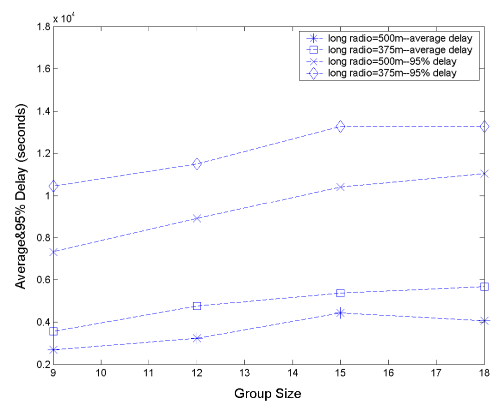

Figure 9: Average/95% Delay with different group sizes

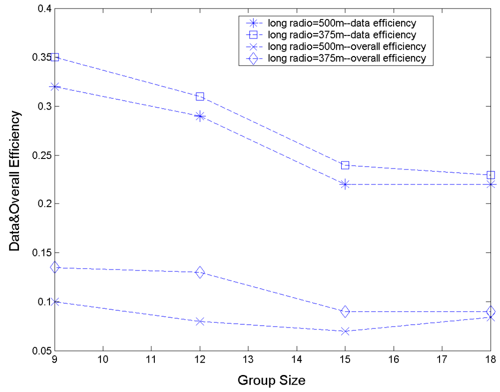

Figure 10: Data/Overall Efficiency with different group sizes

From the plots in Figures 8-10, we see that the delivery ratio is maintained at high values (more than 75%) even with larger group size. The data efficiency drops slightly with larger group size. The overall efficiency drops initially but then improves as the group size grows. We also observe that using a high power transmission range of 375 m will provide slightly smaller delivery ratio but slightly better data/overall efficiency.

4.4 Impact of Traffic Load

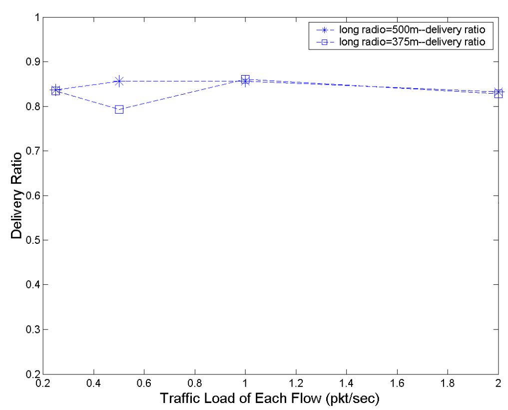

Figure 11: Delivery Ratio at different traffic load

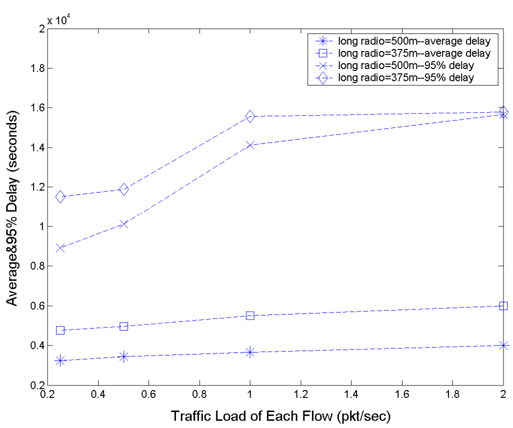

Figure 12: Average/95% delay at different traffic load

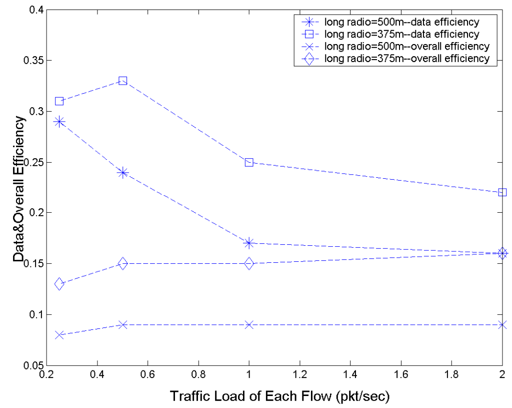

Figure 13: Data/Overall Efficiency with different traffic load

Next, we investigate the impact of traffic load on the performance. As in Section 4.3, we use random waypoint mobility model and an area of 4000x4000 m^2. Figures 11-13 plot the results we obtained when we varied the traffic load. Note that the x-axis is the number of packets sent per second for each flow so the traffic load increases towards the right. From the plots in Figures 11-13, we see that the delivery ratio increases slightly with increasing traffic load. The average and 95% delay increases with increasing traffic load. The data efficiency drops a little with increasing load but the overall efficiency increases slightly with increasing load.

4.5 Comparisons with Other Multicast Approaches

In this section, we compare our CAMR scheme with U-Multicast, DTBR, and OS-Multicast schemes using random waypoint mobility model and a scenario with 25 nodes randomly distributed in 1000x1000 m^2 and a traffic load of 1 bundle every 2 seconds.

Table 5: Comparison between CAMR and others

	Delivery Ratio	Average Delay	Data Efficiency
CAMR	90.5%	21 seconds	0.32
DTBR	74.0%	0.2 seconds	0.42
OS-multicast	48.9%	20 seconds	0.08
U-multicast	42.0%	0.1 seconds	0.46

From the results in Table 5, we see that CAMR achieves the highest delivery ratio with reasonably high data efficiency. The price to pay is the increased average message delivery delay. We note from our results that 80% of the message does have a delivery latency of 10 seconds. The larger average delay is caused by a longer tail in the delivery latency distribution. We expect that the CAMR scheme will outperform all other schemes when the network becomes very sparse. Since store-and-forward approach is not used for real time traffic but to ensure that many messages can be delivered even in scenarios where no end-to-end paths exist, we believe that our CAMR scheme is the best choice for multicast delivery especially when the networks are very sparse e.g. with only 0.5 neighbor within its transmission range. More comparison results will be presented in the journal version of our paper.

5. CONCLUSION

In this paper, we have developed a context-aware adaptive multicast routing scheme for DTNs. Our scheme is flexible and can adapt to different network environments e.g. different node densities, different mobility models. We have evaluated our CAMR scheme and compare it with other existing proposed multicast routing schemes for DTNs. Our results show that the CAMR scheme is more flexible and can provide high delivery ratio with the highest data and overall efficiency especially when the network becomes sparser. For future work, we intend to evaluate the performance of CAMR scheme for multiple multicast sessions. We also intend to integrate CAMR scheme with the message ferrying scheme [20], [21] where special dedicated nodes are used as message ferry and evaluate the integrated scheme in more complex DTN scenarios. Last but not least, we intend to implement our scheme in a DTN testbed for empirical evaluations.

6. ACKNOWLEDGMENTS

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.

7. REFERENCES

[1] K. Fall, "A delay-tolerant network architecture for challenged Internets", In *Proceedings of SIGCOMM*'03, August 2003.

[2] W. Zhao, M. Ammar, and E. Zegura, "Multicasting in delay tolerant networks: semantic models and routing algorithms," in the *Proceeding of Sigcomm Workshop in DTN*, August 2005.

[3] DARPA Disruption Tolerant Networks program http://www.darpa.mil/ato/solicit/dtn/, accessed on Aug. 3rd, 2005.

[4] K. Fall, "Messaging in difficult environments," Intel Research Berkeley, *IRB-TR*-04-019, Dec. 27, 2004.

[5] M. Chuah, L. Cheng, B. Davison, "Enhanced disruption and fault tolerant network architecture for bundle delivery (EDIFY)", Proceedings of IEEE *Globecom*, 2005.

[6] D.B. Johnson and D.A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, edited by T. Imielinski and H. Korth, chapter 5, pp.153-181, Kluwer Academic Publishers, 1996.

[7] Rajeswari Malladi and Dharma P. Agrawal, "Current and future applications of mobile and wireless networks", *Communications of the ACM*, Vol. 45, pp. 144-146, 2002.

[8] 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.

[9] Alberto Cerpa, Jeremy Elson, Deborah Estrin, Lewis Girod, Michael Hamilton, Jerry Zhao, "Habitat monitoring: application driver for wireless communications technology", in the *Proceeding of ACM SIGCOMM Workshop on Data Communications*, April 2001.

[10] S. Jain, K. Fall, R. Patra, "Routing in a delay tolerant networking", in the *Proceedings of SIGCOMM*'04, Aug./Sep. 2004.

[11] J. Moy, "Multicast extensions to OSPF", IETF RFC 1584, 1994.

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

[13] 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.

[14] Sang Ho Bae, Sung-Ju Lee, William Su, and Mario 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

[15] H. Wu, R. Fujimoto, and G. Riley "Analytical models for data dissemination in vehicle to vehicle networks", Proceedings of IEEE VTC Fall 2004.

[16] J. Burgess, B. Gallagher , D. Jensen, B. N. Levine, "MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks", Proceedings of IEEE Infocom, March, 2006.

[17] Q. Ye, L. Cheng, M. Chuah, B. D. Davison, "On-Demand Situation Aware Multicasting in DTNs", Proceedings of IEEE VTC, Spring, 2006.

[18] Y. Wang, S. Jain, M. Martonosi, K. Fall, "Erasure Coding Based Routing for Opportunistic Networks", Proceedings of Sigcomm Workshop in WDTN, Aug, 2005.

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

[20] 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.

[21] M. Chuah, P. Yang, "Message Ferrying Scheme with Differentiated Services", Proceedings of IEEE Milcom, 2005.