Anycast Routing in Delay Tolerant Networks

Yili Gong[1, 2, *], Yongqiang Xiong[3], Qian Zhang[4], Zhensheng Zhang[5], Wenjie Wang[6, *] and Zhiwei Xu[1]
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China[1]
Graduate School of the Chinese Academy of Sciences, Beijing, China[2]
Microsoft Research Asia, Beijing, China[3]
Department of Computer Science, Hong Kong University of Science and Technology, Hong Kong, China[4]
San Diego Research Center, San Diego, CA, USA[5]
EECS Department, University of Michigan, MI, USA[6]
Email: gongyili@ict.ac.cn, yqx@microsoft.com, qianzh@cs.ust.hk, zzhang@ieee.org, wenjiew@eecs.umich.edu and zxu@ict.ac.cn

Abstract — **Anycast routing is very useful for many applications such as resource discovery in Delay Tolerant Networks (DTNs). In this paper, we first analyze the anycast semantics for DTN based on a new DTN model. Then we present a novel metric named EMDDA (Expected Multi-Destination Delay for Anycast) and a corresponding routing algorithm for anycast routing in DTNs. Extensive simulation results show that the proposed EMDDA routing scheme can effectively improve the efficiency of anycast routing in DTNs. It outperforms another algorithm, Minimum Expected Delay (MED) algorithm, by 11.3% on average in term of routing delays and by 19.2% in term of average max queue length.**

Index Terms—**Anycast routing, Routing metrics, Delay Tolerant Network (DTN)**

I. INTRODUCTION

Delay Tolerant Networks (DTNs), as a class of useful but challenging networks, are receiving more and more attention [1][2]. In such networks, no end-to-end contemporaneous path is guaranteed between any two nodes and message[†] delivery can only be fulfilled by leveraging nodes' movement. In this paper, we study the problem of anycasting in DTNs as more and more potential DTN applications need support of anycasting.

Anycast [3] is a service that allows a node to send a message to at least one, and preferably only one, of the members in a group. The idea behind anycast is that a client wants to send packets to any one of several possible servers offering a particular service or application but does not really care any specific one [4]. Anycast can be used to implement resource discovery mechanisms which are powerful buildings block for many distributed systems, including file sharing etc.

Anycast in DTNs means that a node wants to send a message to any one of a destination group and intermediate nodes help to deliver the message by leveraging their mobility when no contemporaneous path exists between the sender node and any node of the destination group. A typical scenario, shown in Fig.

1, is in a park, people cluster to watch some musical performances and they want to share and search music files at the same time. People in different clusters may be disconnected while people or cars moving between clusters can act as carriers to deliver messages. Anycast can be used to find a person who owns a certain file. Moreover, DTN anycast can be used in a disaster rescue field, in which people may want to find a doctor or a fireman without knowing their IDs or accurate locations. Thus efficient anycast service is important for supporting these applications in DTNs.

Though anycast in the Internet and mobile ad hoc networks has been studied extensively in the past, due to the unpredictability of network connectivity and delay, and limited storage capacity, anycast in DTNs is a quite unique and challenging problem. It requires both re-definition of anycast semantics and new routing algorithms. Moreover, in unicast in DTNs, the destination of a message is determined when it is generated, while in anycast, the destination can be any one of a group of nodes. Thus during the routing, both the path to a destination group member and the destination of the anycast message can be changed dynamically according to current mobile device movement situation, correspondingly enlarging the delivery delay.

In this paper, we define three semantics models of anycast in DTNs, namely CM (Current Membership), TIM (Temporal Interval Membership) and TPM (Temporal Point Membership Model), which unambiguously define the intended receivers of a message in the unicast routing. Based on the model, we also propose a novel routing metric named EMDDA (Expected Multi-Destination Delay for Anycast) which utilizes the uncontrolled random moving characteristic of mobile devices. We
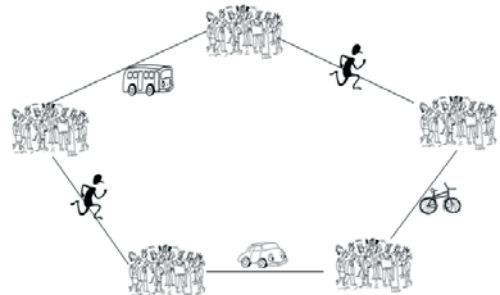

Fig. 1. A typical scenario of DTNs. People clusters and cars, buses or other people moving among the clusters can act as carriers to deliver messages.

---

*This work was done while Yili Gong and Wenjie Wang were visiting students at Microsoft Research Asia.
[†] In this paper, we use term message to represent all kinds of delivered information, including quests, replies, and files, etc.

use a probabilistic approach that characterizes the practical average delay more accurately and develop a corresponding algorithm. The performance of the algorithm is analyzed and compared by simulation which demonstrates the effectiveness and efficiency of the proposed metric. Our simulations show that the EMDDA routing scheme can effectively improve the efficiency of anycast routing in DTNs. It achieves 11.3% lower average routing delays and 19.2% lower average max queue length compared to another algorithm, Minimum Expected Delay (MED).

To the best of our knowledge, this paper is the first to systematically address the anycast routing problem in DTNs and propose the corresponding unicast routing scheme.

The rest of this paper is organized as follows. We first review related research in Section II. Section III presents our analysis of anycast in DTN and our scheme for it. The simulation and performance evaluation are discussed in Section IV. We conclude the paper with Section V.

II. RELATED WORK

There is a rich literature on anycast routing in the Internet and mobile ad hoc networks [5][6][7][8][9]. But these approaches can not be applicable to DTNs since for anycast routing in DTNs it can not assume the connectivity is guaranteed as that in Internet, and the uncertainty of both the path to a destination group member and the destination of the anycast message during the routing makes the problem more challenging.

DTN routing attracts a large amount of efforts in the research community. Vahdat and Becker firstly propose exploiting device mobility to facilitate data routing [10]. They show that with unlimited node buffer size and time, the flooding algorithm guarantees delivery of messages. Tan et al in [11] present a routing framework named SEPR in partially connected networks. By guiding message flow to expected shortest path nodes, their approach reduces unnecessary message copies and increases the message delivering rate.

Zhao et al in [13][14] exploit non-randomness in a node's proactive movement to deliver messages in order to improve data delivery performance in a disconnected network. In our paper we assume that we know the mobility pattern of mobile devices, which is uncontrolled and can't be changed. Zhao et al also propose new semantic models for DTN multicast and develop several multicast routing algorithms with different strategies [16].

Recently, Jain et al have developed and compared several routing algorithms in DTNs [2], including MED (Minimum Expected Delay). With MED, as proposed in [12], the routing table is recomputed each time a contact arrives and messages are exchanged if the topology suggests that a connected node is "closer" than the current node.

However, MED is a routing metric based on the contact summary, for example, the average waiting time, propagation delay and transmission delay. It uses the path with the minimum average delay for all messages with the same source-destination pair. This approach depends on the average information and it doesn't take into account the dynamic changes in the network. As we'll illustrate in Section III, we need consider uncontrolled random moving characteristic of mobile devices and the practical average delay accurately. Furthermore, these routing approaches are designed either for unicast, which is to deliver a message to a determined destination, or for multicast, which is to distribute a message to a group of members. Contrarily, Anycast in DTN aims to deliver a message to any one of a group, usually the nearest one which requires a model to define the precise semantics of anycast in DTNs.

III. EMDDA-BASED ANYCAST IN DTNS

In this section, we first introduce our network model. Then we present the anycast semantics in DTNs. Finally, we develop our routing algorithm based on our anycast semantics. Our algorithm is based on the expected shortest path routing algorithm in which routing decision can be re-made at intermediate nodes according to the movement of mobile devices.
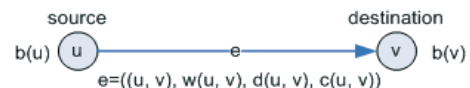

Fig. 2. An edge in a DTN graph, which is characterized by its source and destination nodes, plus a PDF of mobile device leaving (w_e), a moving delay (d_e) and a storage capacity (c_e). $b(u)$ is the storage capacity of node u.

A. Network Model

In this paper, we use a DTN model that is different from the one in [2]. The DTN graph is a directed graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. An edge between node u and v means that there exist some mobile devices moving from the initial node u (source) to the terminal node v (destination) (see Fig. 2). The storage capacity on all nodes and the mobile devices is limited. We assume that every mobile device that moves between the same initial node, u, and terminal node, v, has the same moving speed, thus having the same moving delay, $d(u, v)$, from the source to the destination. The departure time of the mobile devices is a random variable and follows some given probability distribution. $w(u, v)$ is the probability distribution function of the waiting time until a mobile device leaves from the source to the destination. $c(u, v)$ is the storage capacity or buffer size of the mobile device.

We assume nodes in the network are stationary and generate messages. On the other hand, mobile devices don't generate messages themselves. They move from one node to another at will and their mobility can't be controlled from the nodes. Nodes might be disconnected themselves, thus mobile devices can act as carriers to deliver messages for the nodes. Before a mobile device departs from its initial node, the initial node can upload selected messages onto the mobile device. When the mobile device arrives at its terminal node, the terminal node will download the messages.

Fig. 1 illustrates a simple example of our DTN model, in which a single node in the graph represents a cluster of users:

the cluster is relatively stationary and the users in the cluster only move within the cluster. There are some mobile devices, such as shuttle buses, uncontrolled cars or people, moving among the clusters. The exact schedule of the moving devices is unknown, but their moving patterns can be obtained.

Routing in such DTNs is done in a store-and-forward fashion. The routing algorithm determines which mobile device a message should be forwarded to in order for the message to reach its destination as soon as possible.

B. Anycast Semantics

One of the challenges in designing an anycast routing protocol is to maintain the group membership efficiently. Due to the long delivery delay in DTNs, group membership may already change during the delivery of a message, introducing ambiguity in anycast semantics. For example, if a message is intended for group G with member a, b, and c when it is sent, when it arrives, the membership of group G may change to a, d, and e. The anycast routing scheme has to deal with such membership dynamics. It needs a mechanism to distinguish a random group member from the intended receiver of a message. The intended receiver should be clearly defined for a message as group membership changes when nodes join and leave the group. Next we describe three anycast semantic models that allow message senders to explicitly specify the intended receivers of a message.

Current Membership Model To determine the intended receivers of a message, a key element is to explicitly identify the time interval during which the receivers are defined. In the Current Membership (CM) Model, a message should be delivered to a node which is a destination group member when the message arrives at it.

Consider an example of such a model. In a resource discovery system, some index servers, which hold the index information of all resource in the system, are placed in the network according to some performance objective, such as limiting the response time of resource lookup, or satisfying the traffic constraints on each edge. These index servers form an anycast group. If one of these server leaves, or new members join the network, the placement of servers might change in order to meet the response time limit or traffic constraint. Therefore, the receivers of the anycast message may change over time. With the CM model, a message requesting for some resource index should be sent to any of the current index servers.

Temporal Interval Membership Model Our second model is Temporal Interval Membership (TIM) Model. In the TIM model, a message includes an interval that specifies the period during which the group members are defined. For a message in an anycast group G with temporal interval $[t_1, t_2]$, its intended receiver must be a member of group G at any time during the interval.

Using the same resource discovery system example in the CM model, in the TIM model, If the membership interval is [0, 100], the message must be delivered to one node which was an index server all through that period, even if when it arrives at
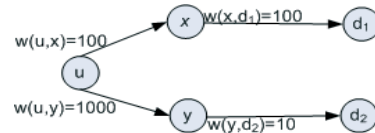

Fig. 3. A simple example of changing route. In this graph, we assume the moving delay of each edge is identical. According to the MED approach, a message should take the path u→x→d_1. But when a mobile device from u to y is leaving, the better choice is the route u→y→d_2.

the node, the node is not an index server anymore.

Temporal Point Membership Model In the third model, Temporal Point Membership (TPM) model, a message also includes a membership interval, similar to the TIM model. For a message for an anycast group G with temporal interval $[t_1, t_2]$, its intended receiver at least should be a member of group G at some time during the interval.

Similarly, using the example of the previous resource discovery system, if the interval of a message is [0, 100], any node which was an index server during that period of time is a desired destination. If the interval is set to [0, +∞], the intended receiver could be anyone of the nodes which were/are an index server since time 0.

The above three models can be used in practical applications according to different context. If the resource index information changes frequently, the CM model is more suitable. Otherwise, the TIM or TPM model is enough.

In the next sections, we just use the CM model to define and explain our anycast routing metric and algorithm. The proposed routing metric and algorithm can be easily applied to the TIM and TPM model.

C. Expected Multi-Destination Delay for Anycast

We propose a metric EMDDA (Expected Multi-Destination Delay for Anycast) which accurately indicates the delay from a node to the nearest member of the destination anycast group. We use Practical Expected Delay (PED) to denote the delay between any two neighboring nodes which takes into account the probability of choosing each neighbor as the next node. EMDDA of a node to an anycast group is the smallest PED among the PEDs from the node to all the destination group members. In the following sections, we will explain how EMDDA is calculated step by step.

Given the network graph $G = (V, E)$, and suppose a message m which should be sent to any member of a group $D_m = \{d_0, d_1, ..., d_{L-1}\}$, $d_i \in V$, $0 \le i \le L-1$, and L is the size of the group. We assume that before a mobile device starts moving, all messages that it should carry can be uploaded as long as the storage of the mobile device is not full.

One simple anycast routing algorithm is to use the sum of the expected waiting time ($E(w(u, v))$) and moving delay ($d(u, v)$) as the weight of each edge, then to calculate the shortest path to each group member and choose the one with the smallest path weight as the anycast destination. We called this Minimum Expected Delay (MED) approach. Let us use MED(s, d) to represent the minimum expected delay between nodes s and d, s, $d \in V$. The value of MED can be calculated through a shortest

path algorithm, such as Dijkstra algorithm.

While a message is waiting for a mobile device to the next node on the path of the minimum expected delay, another mobile device to another destination comes, should the message change its destination and use this mobile device? The answer is yes, if some conditions hold. Fig. 3 shows a simple example of changing route. According to the MED approach, an anycast message heading for a group $\{d_1, d_2\}$ should follow the path $u \rightarrow x \rightarrow d_1$. While the message is waiting a mobile device to node x, a mobile device to node y comes, the message should change its route, take the mobile device to y and arrive at d_2 finally. This is because the leaving time of mobile devices is random and the expected waiting time can't accurately reflect the specific situation that each message meets on its way to its destination. Next, we will find out under what conditions a message should change its route.

We propose a new routing metric called Practical Expected Delay (PED) to characterize the expected delay of taking different paths with corresponding probability between a node pair. For a node c, $Nb(c)$ represents its neighbors. Let $P(c, d, a)$ be the probability that a message at node c to destination node d takes node a as the next step node, $a \in Nb(c)$. Then

$$PED(c,d) = E(w(c,d)) + \sum_{a \in Nb(c)} P(c,d,a) \cdot (d(c,a) + PED(a,d)) \cdot$$

Next, we will calculate the value of $P(c, d, a)$ ($a \in Nb(c)$). If $d(c, a) + MED(a, d) > MED(c, d)$, $P(c, d, a) = 0$, i.e. while waiting, even if a mobile device to node a ready to move, the message won't take it. Let set $A(c, d) = \{a \in Nb(c) \mid (d(c, a) + MED(a, d)) <= MED(c, d)\}$.

$P(c, d, a)$, $a \in A(c, d)$, is equal to the probability that a mobile device to node a is about to leave before any mobile device to any other node.

$$P(c,d,a) = \int_0^{+\infty} f_a(t) \cdot \prod_{\substack{b \in A(c,d) \\ b \neq a}} (1 - F_b(t)) \cdot dt$$

($a \in A(c, d)$). Here $f_a(t)$ is the probability density function of the

---

waiting time before a mobile device is about to leave from c to a, and $F_b(t)$ is the distribution function of the waiting time that a mobile device is leaving for b.

Assume on each edge the departure times of mobile devices follows Poisson distributions with $\lambda = \lambda_0, \lambda_1, ..., \lambda_{p-1}$ respectively. Then

$$E(w(c,d)) = \frac{1}{\sum_{a \in A(c,d)} \lambda_a}$$

$$P(c,d,a) = \int_0^{+\infty} \lambda_a e^{-\lambda_a t} \cdot \prod_{\substack{b \in A(c,d) \\ b \neq a}} e^{-\lambda_b t} \cdot dt = \int_0^{+\infty} \lambda_a e^{-\sum_{b \in A(c,d)} \lambda_b t} \cdot dt = \frac{\lambda_a}{\sum_{b \in A(c,d)} \lambda_b}$$

We extend the metric PED to anycast routing and get a metric called Expected Multi-Destination Delay for Anycast (EMDDA). $EMDDA(s, D) = min(PED(s, d_0), PED(s, d_1), ..., PED(s, d_{L-1}))$, $s \in V$, and D is the destination node set and $D = \{d_0, d_1, ..., d_{L-1}\}$.

The key advantage of the metric EMDDA is that it can reflect the expected delay between a pair of nodes by taking all possible paths into account instead of only the shortest path.

D. Routing algorithm

Our anycast routing algorithm is based on the metric EMDDA, shown in Fig. 4. On a message m arriving at a node s, if the node is an intended receiver of the anycast message, the process stops. Otherwise, the node will calculate the EMDDA from this node to the destination group. When a mobile device is about to leave from node s to node z, if $(d(s, z) + EMDDA(z, D)) < EMDDA(s, D)$, node s will upload message m onto the mobile device, otherwise do nothing.

If mobile devices move infrequently between two nodes, the edge between them will have high expected delay. Thus the shortest path algorithm won't use these edges. However, if the edge is available immediately, its waiting time becomes zero at this moment and it may be a very good choice for routing. One advantage of our routing algorithm is that it can make use of this phenomenon.

IV. PERFORMANCE EVALUATION

In this section, we evaluate the performance improvement EMDDA-based algorithms can achieve over MED-based algorithms when applied to anycast in DTN environments. We have discussed that the following two factors play important roles in the performance of EMDDA-based algorithms, the storage capacity of the mobile devices and the mean inter-arrival time of messages. In this section, we investigate the effect of these two factors on our anycast routing schemes together with several other factors, such as the routing distance.

A. Simulation Setup

In our simulation, we employ Waxman Network Topology Generator [17] to generate a random graph of 100 nodes. In the Waxman generator, nodes are generated by a Poisson process in the plane with scaled Lévesque mean measure. Nodes u and v are connected with probability $P(u, v) = alpha*exp(-d(u, v)/(beta*L))$, where alpha>0, beta<=1, $d(u, v)$ is Euclidean distance, and L is the maximum distance between any two

---

- **On arrival of a message m, whose destination group is $D = \{d_0, d_1, ..., d_{L-1}\}$**

  If local node s is a member of D, then
      The message arrives at its destination;
      The process stops.
  Else /* Calculate $EMDDA(s, D)$ */
      For each $d_i \in D$ do
        Calculate $MED(s, d_i)$;
        Set $A(s, d_i) = \{a \in Nb(s) \mid (d(s, a) + MED(a, d_i)) <= MED(s, d_i)\}$;
        $PED(s, d_i) = E(w(s, d_i)) + \sum_{a \in A(s,d_i)} P(s, d_i, a) \cdot (d(s, a) + PED(a, d_i))$
      End For
      $EMDDA(s, D) = min(PED(s, d_0), PED(s, d_1), ..., PED(s, d_{L-1}))$;
  End if

- **On a mobile device leaving for node z**

  If $(d(s, z) + EMDDA(z, D)) < EMDDA(s, D)$, then
      Upload the message m onto the mobile device
  Else /* if taking the mobile device, the expected anycast delay would not be shortened */
    Do nothing.
  End if

Fig. 4. The EMDDA-based anycast routing algorithm in DTNs.

nodes. In our simulation we set lambda to 0.6, alpha to 0.4, beta to 0.2. The generated graph has an average degree of 8.14. We have also conducted simulations with other settings, and the qualitative results are consistent with the results we present here.

We want to mention that in extreme cases where nodes in the topology all have small degree (degree of two, for example), the improvement shown by EMDDA diminishes since there are very limited alternatives for EMDDA to select during the routing process.

To simulate the behaviors of DTNs, we assume the communication between nodes in the topology is carried out by mobile devices. For each edge generated by Waxman generator, we replace it with a mobile devices acting as ferries. The leaving time of mobile devices on each edge follows Poisson distributions and the mean interval time is selected randomly from 600 to 6,000 seconds. The moving delay, or single-hop delay, on each edge is a number between 60 and 600 seconds, which is in proportion to the distance between the nodes. We assume that the storage capacities of mobile devices are the same and they vary from 300 to 800 messages.

We consider only anycast traffic in our simulations. For each anycast *session*, we randomly pick two nodes as the anycast destination group and another node as the anycast source. Each source sends 200 messages to the destination group. The messages are generated at each sender according to a Poisson process with mean inter-arrival times varying from 75 to 200 seconds.
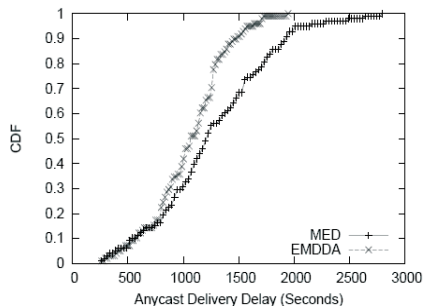

Fig. 5. CDF of anycast delivery delay (ADD). Here the mean message inter-arrival time is 100 seconds and the mobile device storage capacity is 300 messages.
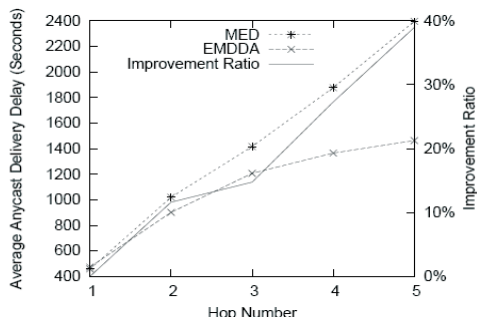

Fig. 6. Relationship of average anycast delivery delay (AADD) with hop number and improvement ratio of delay. Here, the mean message inter-arrival time is 100 seconds and the mobile device storage capacity is 300 messages.

B. Performance Metric

The primary performance metric we employ is named *Any-*

cast Delivery Delay (ADD). The ADD of a message is defined as the time it takes to route this message from its sender to any node in its anycast destination group. Based on ADD, we defined *Average Anycast Delivering Delay* (AADD), which is the average ADD of all possible anycast sessions, i.e., from each sender to any anycast destination groups.

The other metric we use is *average max queue length*, which is the average of the max queue lengths on all the nodes. Messages will be queued on intermediate nodes if the mobile devices to the next hop are not available or the mobile devices do not have sufficient capacity to carry them. The queue length can be large if a sub-optimal route is selected, where the frequency of the mobile device is low, the capacity of mobile device is small, or a large number of messages are routed through the same route. Messages may be dropped if the queue length exceeds the buffer size allocated by intermediate nodes. Average max queue length evaluates buffer size, or space overhead, required on intermediate nodes.
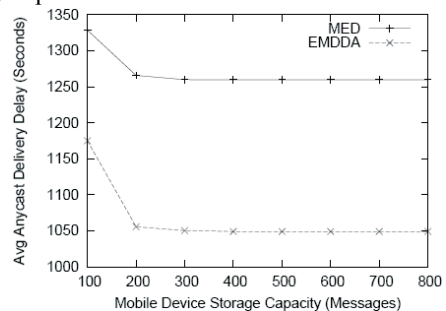

Fig. 7. Average anycast delivery delay (AADD) over mobile device storage capacity. Here the mean message inter-arrival time is 100 seconds.
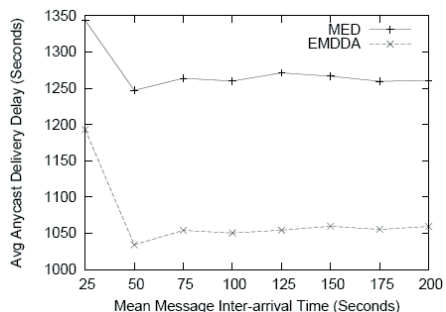

Fig. 8. Average anycast delivery delay (AADD) over mean message inter-arrival time. Here the mobile device storage capacity is 300 messages.

C. Anycast Delivering Delay and Average Anycast Delivering Delay

We have evaluated EMDDA and MED anycast routing algorithms under various traffic rates and mobile device storage capacities. Here we only present the results with representative performance.

We first compare the cumulated distribution function (CDF) of ADD between EMDDA and MED algorithms in Fig. 5. We can see that about 90% of the S-D pairs in EMDDA have ADDs below 1,500 seconds while the number for MED is only 65%. EMDDA can effectively reduce the delay in anycast routing. In terms of the maximum ADD, the number for EMDDA is 1,945.2 seconds, which is 30.4% lower than that of MED, 2,795.6 seconds.

In Fig. 6, we study the effect of routing path length on the AADD performance. We expect that the improvement of EMDDA over MED will increase as the hop number increases since EMDDA can route messages to a closer destination intelligently along the route. In other words, more hops provide more chances for EMDDA to switch to a shorter path other than the original MED path. Fig. 6 confirms our expectation. We can see that AADD improvement achieved by EMDDA---indicated by the bold line---increases to as high as 38.9% as the number of hops increases from one to five. For routing paths with only one hop, EMDDA and MED perform almost the same, while as the hop number increases, EMDDA increasingly outperforms MED.

In Fig. 7 and Fig. 8 we show respectively the effect of mobile device capacity and message inter-arrival time. For both EMDDA and MED, increasing storage capacity will reduce the AADD because mobile devices can carry more messages in one delivery and consequently reduce the queuing delay in intermediate nodes. Similarly, long message inter-arrival time will also decrease the AADD. Fig. 7 shows that EMDDA can reduce the average delay by 11.3% on average compared to MED.

D. Average Max Queue Length

The comparison of average max queue length with mean message inter-arrival time is shown in Fig. 9. As the mean message inter-arrival time increases, for both MED and EMDDA, the average max queue length decreases. However, the average max queue length of EMDDA is always smaller than that of MED. The maximum improvement EMDDA gains over MED is 22.6% with the average improvement being 19.2%. This set of simulation shows that EMDDA can effectively reduce space overhead required on the intermediate nodes.
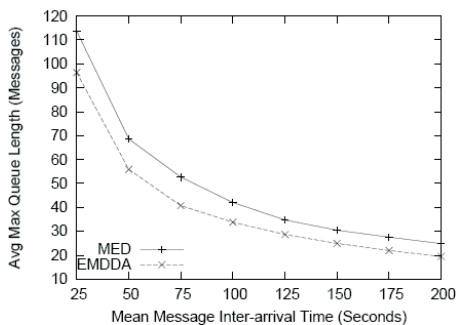

Fig. 9. Average max queue length of all nodes with mean message inter-arrival time. Here the mobile device storage is 300 messages.

V. CONCLUSION

In this paper, we analyze the anycast semantics for DTN based on a new DTN model. Three anycast semantics models are proposed, which unambiguously define the intended receivers of messages and have various applications in DTN environments. We then present a novel routing metric named EMDDA (Expected Multi-Destination Delay for Anycast) and a corresponding routing algorithm for anycast routing based on

the CM model. The main advantage of EMDDA is that it depicts the practical expected delay for anycast more accurately than metric MED (Minimum Expected Delay). Our simulation results show that EMDDA can reduce the average delay by 11.3% on average compared to MED and reduce the amount of buffer/space overhead by 19.2% on average.

The anycast routing algorithm proposed in this paper does not consider network traffic during its routing selection. As one of the interesting future works, we can extend our anycast routing scheme to incorporate both node storage constraint and network traffic dynamics.

REFERENCES

[1] K. Fall. A delay-tolerant network architecture for challenged internets. In Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, 2003, Karlsruhe, Germany, pp 27 – 34.

[2] S. Jain, K. Fall and R. Patra. Routing in a delay tolerant network. In Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications, 2004, Portland, Oregon, USA, pp: 145 – 158.

[3] C. Partridge, T. Mendez and W. Milliken. Host anycasting service. In RFC 1546, Nov. 1993.

[4] C. Metz. IP anycast point-to-(any) point communication. Internet Computing, IEEE, Volume 6, Issue 2, March-April 2002, pp: 94 – 98.

[5] D. Katabi and J. Wroclawski. A framework for scalable global ip-anycast (GIA). In Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, Stockholm, Sweden, 2000, pp: 3 – 15.

[6] S. Bhattachargee, M. Ammar, E. Zegura, N. Shah and Z. Fei. Application layer anycasting. In Proc. Infocom'97, 1997.

[7] V. D. Park, J. P. Macker. Anycast routing for mobile networking. In Proceedings of the IEEE Military Communications Conference, 1999, Volume 1, Page(s):1 - 5 vol.1.

[8] R. R. Choudhury and N. H. Vaidya. MAC-layer anycasting in ad hoc networks. ACM SIGCOMM Computer Communication Review, Volume 34, Issue 1 (Jan. 2004), pp: 75 – 80.

[9] C. Intanagonwiwat and D. D. Lucia. The sink-based anycast routing Protocol for Wireless Sensor Networks. Technical Report 99-698, Computer Science Department, University of Southern California. January, 1999.

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

[11] K. Tan, Q. Zhang, W. Zhu. Shortest path routing in partially connected ad hoc networks. In Proceedings of the IEEE Global Telecommunications Conference, 2003 (GLOBECOM '03), Volume 2, 1-5 Dec. 2003 Page(s):1038 – 1042.

[12] E. P. C. Jones, L. Li, and P. A. S. Ward. Practical routing in delay-tolerant networks. In Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking (WDTN '05), August 22-26, 2005, Philadelphia, PA, USA.

[13] W. Zhao, M. Ammar and E. Zegura. Multicasting in delay tolerant networks: semantic models and routing algorithm. In Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking (WDTN '05), August 22-26, 2005, Philadelphia, PA, USA.

[14] W. Zhao and M. Ammar. Message Ferrying: Proactive Routing in Highly-partitioned Wireless Ad Hoc Networks. In Proceedings of the Ninth IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS'03).

[15] W. Zhao, M. Ammar and E. Zegura. A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks. In Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing, May 2004, Roppongi Hills, Tokyo, Japan, Pages: 187 - 198.

[16] W. Zhao, M. Ammar and E. Zegura. Controlling the Mobility of Multiple Data Transport Ferries in a Delay-Tolerant Network. In Proceedings of the IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies, 13-17 March 2005.

[17] Ingemar Kaj, Raimundas Gaigalas. http://www.math.uu.se/~ikaj/courses/matlab/.