Dynamic Quota-Based Routing in Delay-Tolerant Networks

Shou-Chih Lo and Wei-Rong Liou
Department of Computer Science and Information Engineering
National Dong Hwa University
Hualien 974, Taiwan, ROC
sclo@mail.ndhu.edu.tw

Abstract—**Delay-Tolerant Networks (DTNs) are special types of network environments that are subject to delays and disruptions. Traditional end-to-end routing fails in DTNs due to intermittent connections. Quota-based routing shows good performance in DTNs; however, this routing suffers from a common problem in setting quotas. In this paper, we propose a new routing scheme that can dynamically adjust quota values according to network loads. We evaluate the proposed routing scheme for social contact networks in terms of different performance aspects. Experimental results show the superiority of this routing scheme over other existing ones.**

Keywords-DTN; Routing; Buffering; Store-and-Forward

I. INTRODUCTION

Traditional Internet protocols malfunction in challenged networks such as interplanetary and deep-space networks in which communications are subject to delays and disruptions [1]. These challenged networks are generalized as Delay-Tolerant Networks (DTNs), so named by Kevin Fall in 2002 [2]. A DTN has characteristic properties: high and variable latency, low data rate, frequent disconnection, and high error rate. Node mobility contributes greatly to these characteristics. A *store-and-forward* mechanism is commonly used to transfer data in a DTN. A node can store and carry data in its own buffer, and forward these data to other connected nodes when they are available.

Many application domains using DTNs have been studied. For example, the IPN (Interplanetary Internet) project [3] studies outer-space communications. The ZebraNet project [4] uses sensors to study interactions between zebras. The DakNet project [5] offers communication for remote regions. The Haggle project [6] uses hand-held devices to exchange information among people.

DTN related standards are studied by the Delay Tolerant Networking Research Group (DTNRG) [7]. DTN architecture was introduced in RFC 4838 [8], with a new bundle layer [9] added between the application layer and the transport layer. The bundle layer implements the store-and-forward mechanism. Data packets passing through the bundle layer are grouped into basic units called *bundles* or *messages*.

Due to intermittent connections in a DTN, a stable end-to-end routing path is hard to maintain. The task of routing a message to a destination node in such an environment becomes challenging. Most DTN routing protocols rely on the concept of message replication and path diversity, which generates many message copies to be routed individually along different paths. This feature of routing diversity can greatly increase the message delivery ratio, but will also cause a negative effect if too many message copies are generated.

Quota-based routing [10][11][12], which can limit the number of message copies by specifying a quota presents good performance in DTNs. However, this kind of routing has a weakness in the setting of quota values. A fixed setting scheme lacks the flexibility to react to the change of network conditions. In this paper, we propose a dynamic setting scheme by observing the local network condition.

We first introduce a new metric called *contact density* to evaluate the degree of activity of a node. A node with high activity is given heavy responsibility to spread message copies. Secondly, we propose a mechanism to dynamically adjust the quota associated with a message by measuring the buffer occupancy of neighboring nodes. Finally, we evaluate performance using a real trace of contact events happening in an academic social activity.

The remainder of this paper is organized as follows. Section II gives a brief survey of DTN routing protocols. The proposed dynamic routing is presented in Section III. Performance comparisons are presented in Section VI. Finally, we make some concluding remarks in Section V.

II. RELATED WORK

DTN routing has been widely discussed in recent years. The routing operation can be divided into three main stages. First, when two nodes encounter each other, meta-data including information for buffer management and routing decisions are exchanged. Second, these two nodes refresh data in their own buffers and routing tables. Third, they determine what messages in the buffer need to be copied or forwarded to the peer node, and in what order these messages are transmitted.

Most research efforts in DTN routing focus on two things: buffer management and routing decisions. The buffer management is focused on the arrangement of message transmission order and message drop order, which can be accomplished by sorting messages in a buffer according to certain information indexes such as message received time, message remaining life time, and message delivery hop.

The routing decision aims to determine how to replicate and forward message copies to suitable nodes that are evaluated with regard to certain metrics such as contact frequency and contact waiting time. DTN routing protocols can be roughly classified into three categories according to how many copies

This research was partially supported by National Science Council of the Republic of China under Contract No. NSC 100-2221-E-259-020.

978-1-4673-0990-5/12/$31.00 ©2012 IEEE

of the same message are created: *forwarding*, *quota-replication* (or *quota-based*), and *flooding*.

In a forwarding scheme, a single-copy message is forwarded through successive intermediate nodes to the destination. A quota-replication scheme creates a limited number of message copies as specified by a quota. A flooding scheme creates an extremely large number of message copies and spreads them into the whole network. We refer interested readers to our survey paper [13] for more detailed comparisons of these three routing schemes.

Forwarding protocols such as MEED [14] and SimBet [15] though save buffer space in each node, they present low delivery ratio unless contact events are frequent in the network. In flooding protocols such as Epidemic [16], MaxProp [17], and Prophet [18], a large volume of message copies may exhaust the buffer space of each node, and cause the dropping of some message copies. Flooding protocols have good performance only when the size of buffer space is sufficient.

Quota-replication routing limits the maximum number of copies of the same message by setting an initial quota. The setting of the quota is a tradeoff between resource consumption and message deliverability, and hence is a challenge. Quota-replication protocols such as Spray&Wait [10], EBR [11], and SARP [12] show better performances than the other two routing categories.

In a quota-replication scheme, each message generated from a source is associated with an initial quota. A node can conditionally duplicate a message in the buffer, and forward the duplicated message to another encountered node. Moreover, the duplicated message is allocated part of the quota of the original message based on a *quota allocation function*. Any message whose quota has been reached and cannot be allocated any more is therefore transmitted to the destination by direct contact.

Spray&Wait uses a binary quota allocation function such that a duplicated message gets a half of the quota from the original message. EBR maintains an encounter value for each node, which is the average number of encounters with other nodes during an observation period. The quota allocation function is based on the ratio of the encounter values of two contact nodes. SARP behaves like EBR, but uses the encounter value with the message destination. Moreover, SARP counts the number of encounters between two nodes in a new way. Contacts with a short duration and a long duration would respectively contribute zero and more than one to encounter times.

The message quota is gradually decreased according to the given quota allocation function in the above protocols. The initial setting of the message quota becomes important for getting good routing performance. Typically, a large or small quota is preferred when the network load is light or heavy, respectively. However, the network load may be dynamically changed over time. This motivates us to develop a dynamic quota-based routing protocol that can properly increase or decrease the message quota based on the observation of the current network condition.

III. DYNAMIC QUOTA-BASED ROUTING

Our proposed *Dynamic Quota-Based Routing* (DQBR) contains two main parts: a new quota allocation function and a dynamic quota adjustment mechanism. When two nodes encounter each other, the message holding node may increase or decrease the message quota using the dynamic quota adjustment mechanism. Then, this node duplicates a message copy, with its quota being set by the new quota allocation function.

A. Dynamic Quota Adjustment

A good quota adjustment mechanism should consider global network condition. However, this requires a lot of control messages to be exchanged extensively. Here, we observe the buffer occupancies of local nodes to estimate local network condition. A message becomes hard to be delivered to its destination for two possible reasons. One is that the network becomes congested such that the message is dropped by an intermediate node, as a consequence of the saturation of buffer space. The other is that the network becomes inactive such that the message is stuck at an intermediate node without any encounters with other nodes. The buffer space of this node would be eventually saturated as a result of many stuck messages.

The *Buffer Occupancy* (*BO*) of a node becomes high in the above two situations. Hence, we observe the current *BO* and even predict the future *BO* of a node. In the following, we introduce a method to estimate the future *BO*. The *BO* with values between 0 and 1 indicates how much of the buffer has been occupied. A node periodically examines the current *BO* (BO_{current}) and uses Eq. (1) to predict the new *BO* (BO_{new}). BO_{old} in the equation is the previously estimated *BO*.

$$BO_{new} = \alpha \times BO_{old} + (1-\alpha) \times BO_{current} \tag{1}$$

When a node encounters another node, they exchange their BO_{new} values. In a wireless communication environment, a node may contact several nodes simultaneously within its communication range. These nodes become neighboring nodes of this node. We use a table BOT (*BO* Table) to record the BO_{new} values of current neighboring nodes. Then, we average these values in the BOT to get an *Average BO* (*ABO*).

This *ABO* is used to indicate the local network condition of a node. When *ABO* is high, the surrounding network of a node is assumed to be congested or inactive. Here, we use a threshold-based method as per Eq. (2) to dynamically adjust the message quota held by a node. Suppose that Q_{current} denotes the current quota associated with a message and Q_{new} is the new quota after the adjustment. When *ABO* is no greater than 0.2, the network is assumed to be lightly loaded, and hence we increase the number of message copies by increasing the message quota by one. When *ABO* is no less than 0.8, we exponentially decrease the message quota. In the other cases, the message quota remains unchanged.

$$Q_{new} = \begin{cases} Q_{current} + 1, if\ ABO \leq 0.2 \\ \lceil Q_{current} \times e^{-Q_{current}/CN} \rceil, if\ ABO \geq 0.8 \\ Q_{current}, Otherwise \end{cases} \tag{2}$$

In decreasing the quota, *CN* denotes the average number of distinct nodes encountered by a node within a time window. The message quota is greatly decreased as *CN* is small, since this node has a smaller number of contact nodes that help to spread the message. The *CN* is computed by each node as per Eq. (3). CN_{new} and CN_{old} are the new and the old estimated values respectively. CN_{current} is the current accumulated number of contact nodes. CN_{current} is increased by one when a node meets another node that is first encountered within the current time window. The time window will slide such that CN_{current} is reset to zero at each beginning of a new time window.

$$CN_{new} = \alpha \times CN_{old} + (1-\alpha) \times CN_{current} \tag{3}$$

B. Quota Allocation Function

To properly measure the contact degree of a node with other nodes, we introduce a new metric called *Contact Density* (*CD*) as shown in Eq. (4).

$$CD = \frac{1}{WT_1} + \frac{1}{WT_2} + \cdots + \frac{1}{WT_k} \tag{4}$$

WT_i is an average contact waiting time for node i. In other words, this value indicates the average time of a node waiting for a contact with another node. The *WT* value is computed by recording the connection duration and the disconnection duration for each contact. The more detailed computation is given in [14]. The reciprocal of *WT* to some degree indicates the contact frequency with a node. Therefore, we sum up these reciprocals for all the nodes that have ever been contacted by a certain node. To avoid keeping too many *WT* values, some only occasionally contacted nodes are ignored. That is, we prune those *WT* values that are greater than a certain threshold.

After obtaining this *CD* value for each node, we introduce our quota allocation function which is used in each contact event. Suppose that nodes i and j encounter each other, and their *CD* values are denoted by CD_i and CD_j. If node i has a message with quota Q_{i_before} and would like to forward a message copy with quota Q_j to node j, the corresponding message quotas are computed as per Eq. (5). After this message forwarding, node i updates its own message quota from Q_{i_before} to Q_{i_after}. As can be seen, our quota allocation function is directly proportional to the ratio of the *CD* values of these two nodes.

$$\begin{cases} Q_j = \left\lfloor Q_{i\_before} \times \frac{CD_j}{CD_i + CD_j} \right\rfloor \\ Q_{i\_after} = Q_{i\_before} - Q_j \end{cases} \tag{5}$$

C. Message Routing Algorithm

The routing algorithm of DQBR is given in **DQBRRouting** which is performed by all nodes. At first, each node periodically maintains two information items. If a node encounters another node, four meta-data items are exchanged between them. Assume that each generated message in the network has a unique global ID (e.g., the combination form of the source node ID, the destination node ID, and a sequence number). The *m-list* summarizes the content of one buffer by listing message IDs within the buffer. By comparing two m-list records, we can avoid transmitting redundant messages between two nodes.

In quota-replication routing, when a message copy reaches the destination, other message copies in the network become garbage. One common technique to clean these garbage messages is to exchange the *i-list* [19]. The i-list records the IDs of messages that are known to have reached their destinations. When a destination node successfully receives a message, this node adds a new record for this message into its i-list. Two contact nodes will exchange and merge their i-list records. To avoid a long i-list, each record in the i-list can be associated with a timestamp such that an old record can be removed by setting an expired time.

After keeping and refreshing some information, we sort the messages in the buffer to determine their transmission order. Here we use a sorting policy which is different from traditional ones such as FIFO (First-In-Fist-Out) and random. Suppose that we annotate each message in the buffer with two information items: received time (entering time into a buffer) and hop count (number of hops taken from the source). Then we sort the messages by the product value of these two information items from small to large. To break a tie, the message ID is considered. We give low transmission priority to a message with a large product value, since this message has travelled the network for a long distance and for a long time. Moreover, when the buffer cannot accommodate a newly received message, some messages with large product values will be dropped.

Next, we examine each message, in order, in the buffer. Those messages with quotas greater than 1 have the opportunity to be copied to another node. Before performing the quota allocation, the quota value is adjusted by our proposed method.

DQBRRouting
Begin
  Periodically update *BO* and *CN* for every time window.
  If an encounter event with any node *j* happens,
    Exchange meta-data (*m-list*, *i-list*, *CD*, and *BO*) with node *j*.
    Update the received *BO* from node *j* into the local BOT.
    Record this contact event to support for the computing of *CN* and *CD*.
    Remove messages that are indicated in the i-list of node *j* from the local buffer.
    Order the messages in the local buffer according to our proposed sorting policy.
    For each message *m* with quota *Q* in the local buffer from the head to the end,
      If message *m* is in the m-list of node *j*,
        Ignore this message.
      Else if the destination of message *m* is node *j*,
        Copy message *m* to node *j* and remove message *m* from the local buffer.
      Else,
        Update $Q = \begin{cases} Q+1, if\ ABO \le 0.2 \\ \lceil Q \times e^{-Q/CN} \rceil, if\ ABO \ge 0.8 \end{cases}$
        If $Q > 1$,
          Compute $Q_j = \left\lfloor Q \times \frac{CD_j}{CD_i + CD_j} \right\rfloor$.
          If $Q_j > 0$,
            Copy message *m* with quota Q_j to node *j* and update Q as $Q - Q_j$.
          End if
        End if
      End if
    End for
  End if
End.

IV. Performance Evaluation

We evaluate routing performance using the Opportunistic Network Environment (ONE) simulator [20]. One trace file which records contact events during the 2005 INFOCOM conference is used. To capture purely social contacts, we have removed all external nodes from the file.

Three cost metrics are used in our evaluation, as below:

- *Delivery ratio*: ratio of the total number of messages received at destinations to the total number of messages sent from sources.
- *Average relay cost*: ratio of the total message relay count to the total number of messages received at destinations. The relay count is increased by one as any message is forwarded or copied from one node to another node in the network.
- *Effective latency*: ratio of the average end-to-end delay for a message from the source to the destination to the delivery ratio.

The effective latency above is different from the conventional end-to-end delay, and can clearly justify a protocol with short end-to-end-delay but with low delivery ratio.

A. Simulation Model

The parameter settings of our experiments are listed in Table 1. Fixed-size messages are constantly generated per interval time during a generation period. Sources and destinations of these messages are randomly selected from the network nodes. Four DTN routing protocols are compared with DQBR: MaxProp, Prophet, Spray&Wait (abbreviated as SnW), and EBR.

Table 1. Parameter Settings.

Parameter	Value
Simulation time	276000 s
Number of nodes	41
Buffer size	1 MB, 5 MB, 10 MB (default), 15 MB, 20 MB
Transmission rate	250 kBps
Message size	500 kB
Message interval	300 s, 250 s, 200 s, 150 s, 100 s (default)
Message generation period	20000 s to 130000 s
Initial quota	4, 8, 12 (default), 16, 20
Time window	1000 s, 2000 s, 3000 s (default), 4000 s, 5000 s
α	0.8

B. Simulation Results

In the first set of experiments, we change the interval time of generating messages. The results are shown in Figs 1-3. As can be seen, DQBR has the highest delivery ratio among all routing protocols. Moreover, the delivery ratio of DQBR is slightly more stable than the others in response to the change of message intervals. This indicates that DQBR can properly adapt itself to different network conditions. The other protocols are significantly degraded in terms of delivery ratio under a heavily loaded condition with a short message interval, because there are too many message copies that cause message dropping in limited buffers.

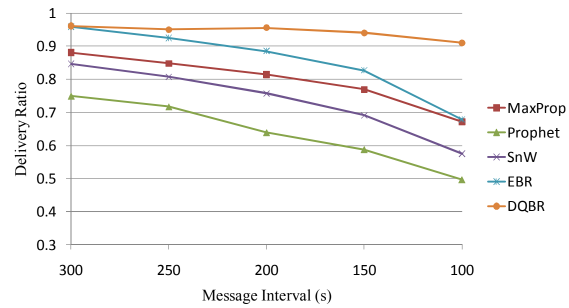

Fig. 1. Delivery Ratio vs. Message Interval.

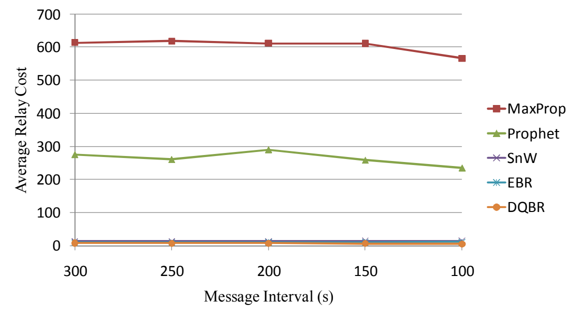

Fig. 2. Average Relay Cost vs. Message Interval.

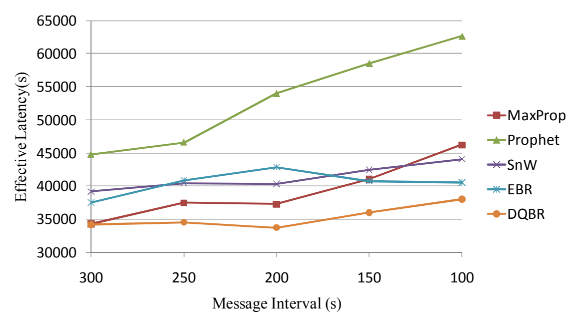

Fig. 3. Effetive Latency vs. Message Interval.

Flooding protocols MaxProp and Prophet have higher relay costs than the quota-replication protocols, since larger numbers of message copies are generated during the message routing stage. DQBR has the lowest relay cost, which is partially contributed to by its high delivery ratio.

DQBR also has the lowest effective latency, and this justifies the usefulness of contact density. We allocate higher message quotas to more active nodes, and this helps to quickly propagate messages to their destinations. MaxProp has a low effective latency, since parts of a large amount of message copies can be delivered via shortest paths to their destinations.

In the second set of experiments, we change the buffer size of each node. The results are shown in Figs. 4~6. When the buffer size is 1 MB, all protocols have similarly low delivery ratios due to too much message dropping. As the buffer size gets larger, the delivery ratio is increased. DQBR quickly reaches a higher delivery ratio than the other protocols. Therefore, our proposed protocol utilizes the buffer space more efficiently.

Flooding protocols experience high relay cost, as before. DQBR still provides good performance in terms of effective latency, except for the condition of 1 MB. The reason for this is that we decrease message quotas due to high buffer occupancy,

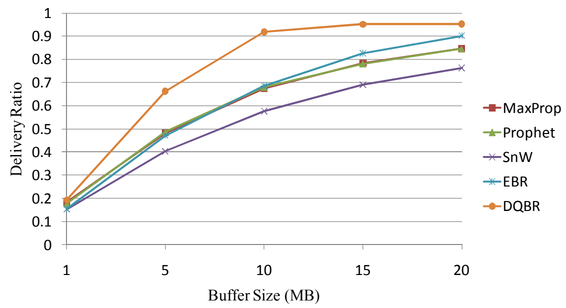

Fig. 4. Delivery Ratio vs. Buffer Size.

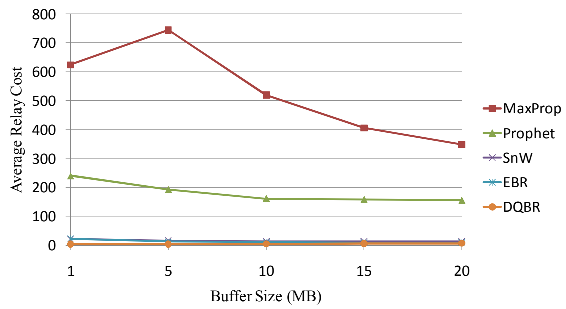

Fig. 5. Average Relay Cost vs. Buffer Size.

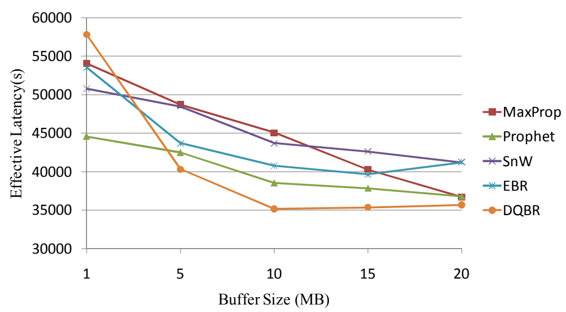

Fig. 6. Effective Latency vs. Buffer Size.

and this causes a smaller number of routing paths to be examined.

In Table 2, we verify whether or not the settings of the initial quota and the time window size (used in the *BO* and *CN* computation) have significant effects on delivery ratio. As can be observed, DQBR is less sensitive than other protocols to these values, in terms of delivery ratio.

Table 2. Parameter Effects

Quota/ Delivery ratio	4/ 0.9	8/ 0.91	12/ 0.91	16/ 0.9	20/ 0.89
Window/ Delivery ratio	1000/ 0.91	2000/ 0.91	3000/ 0.92	4000/ 0.91	5000/ 0.92

V. CONCLUSIONS

Quota-based routing has the excellent feature of limiting the number of message copies, and hence can control network loads in DTNs. However, existing protocols fixedly set message quotas so as to not to provide a good fit for any network condition. In this paper, we propose an enhanced mechanism to dynamically adjust quota values by observing local buffer occupancy. Moreover, we use a new metric, contact density, to identify suitable message relay nodes.

In our experimental results, we have shown the superiority of our routing protocol in terms of different cost metrics. The most important thing is that our routing protocol is less sensitive to the setting of the initial message quota, and hence is simple and robust. In the future, we will study a cooperative method among neighboring nodes to reduce severe traffic congestion in DTNs.

REFERENCES

[1] A. McMahon and S. Farrell, "Delay- and Disruption-Tolerant Networking," *IEEE Internet Computing*, vol. 13, no. 6, pp. 82-87, Nov. 2009.

[2] K. Fall, "A Delay-Tolerant Network Architecture for Challenged Internets, " *in Proc. ACM SIGCOMM Conf.*, pp. 27-34, Aug. 2003.

[3] IPN Special Interest Group (IPNSIG), http://www.ipnsig.org/

[4] The ZebraNet Wildlife Tracker, http://www.princeton.edu/~mrm/zebranet.html

[5] A. Pentland, R. Fletcher, and A. Hasson, "DakNet: Rethinking Connectivity in Developing Nations," *Computer*, vol. 37, no. 1, pp. 78-83, Jan. 2004.

[6] Haggle, http://haggleproject.org/

[7] Delay-Tolerant Networking Research Group (DTNRG), http://www.dtnrg.org

[8] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-Tolerant Networking Architecture," IETF RFC 4838, Apr. 2007.

[9] K. Scott and S. Burleigh, "Bundle Protocol Specification," IETF RFC 5050, Nov. 2007.

[10] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks," *in Proc. ACM SIGCOMM Workshop*, pp. 252-259, Aug. 2005.

[11] S. C. Nelson, M. Bakht, and R. Kravets, "Encounter-Based Routing in DTNs," *in Proc. IEEE INFOCOM Conf.*, pp. 846-854, Apr. 2009.

[12] A. Elwhishi and P. H. Ho, "SARP - A Novel Multi-Copy Routing Protocol for Intermittently Connected Mobile Networks," *in Proc. IEEE GLOBECOM Conf.*, pp. 1-7, Nov. 2009.

[13] S. C. Lo, M. H. Chiang, J. H. Liou, and J. S. Gao, "Routing and Buffering Strategies in Delay-Tolerant Networks: Survey and Evaluation," *in Proc*. IEEE ICPP Workshop, Sept. 2011.

[14] E. P. C. Jones, L. Li, and P. A. S. Ward, "Practical Routing in Delay-Tolerant Networks," *IEEE Trans. Mobile Computing*, vol. 6, no. 8, pp. 943-959, Aug. 2007.

[15] E. M. Daly and M. Haahr, "Social Network Analysis for Routing in Disconnected Delay-Tolerant MANETs," *in Proc. ACM MobiHoc Conf.*, pp. 32-40, Sept. 2007.

[16] A. Vahdat and D. Becker, "Epidemic Routing for Partially-Connected Ad Hoc Networks," Duke University, Technical Report CS-200006, Apr. 2000.

[17] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks," *in Proc. IEEE INFOCOM Conf.*, pp. 1-11, Apr. 2006.

[18] A. Lindgren, A. Doria, and O. Scheln. "Probabilistic Routing in Intermittently Connected Networks," *LNCS*, Springer, vol. 3126, pp. 239-254, 2004.

[19] P. Mundur, M. Seligman, and G. Lee, "Epidemic Routing with Immunity in Delay Tolerant Networks," *in Proc. IEEE MILCOM Conf.*, pp. 1-7, Nov. 2008.

[20] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE Simulator for DTN Protocol Evaluation," *in Proc. Intl. Conf. Simulation Tools and Techniques*, Mar. 2009.