2011 International Conference on Parallel Processing Workshops

Routing and Buffering Strategies in Delay-Tolerant Networks: Survey and Evaluation

Shou-Chih Lo Min-Hua Chiang Jhan-Hua Liou Jhih-Siao Gao
Department of Computer Science and Information Engineering
National Dong Hwa University
Hualien 974, Taiwan
sclo@mail.ndhu.edu.tw

***Abstract*—Delay Tolerant Networks (DTNs) have attracted considerable attention in recent years. This kind of network works in communication environments subject to delays and disruptions. Traditional end-to-end routing fails in DTNs due to intermittent connections. A variety of routing strategies for DTNs have been proposed in the past. In this paper, we present a survey of these strategies and provide a classification. Moreover, we evaluate their use in social contact networks. Buffer management strategies are also needed to support routing operations. The technical issue is to design sorting policies that determine the transmission and drop order of messages in the buffer. We identify several sorting indexes and evaluate their performance.**

Keywords-Delay-Tolerant Network; Replication; Forwarding; Routing; Buffer Management

I. INTRODUCTION

The Internet provides reliable communications by relying on the underlying TCP/IP protocol which sets up an end-to-end data path between a source and a destination. This communication paradigm is not suitable to challenged networks such as interplanetary and deep-space networks in which communications are subject to delays and disruptions [1]. Maintaining a stable data path in spite of intermittent connections is difficult.

These challenged networks are generalized as Delay Tolerant Networks (DTNs), so named by Kevin Fall in 2002 [2]. The DTN has characteristic properties: high and variable latency, low data rate, frequent disconnection, and high error rate. Node mobility contributes greatly to these problems. A store-and-forward mechanism is commonly used to transfer data in a DTN. A node will store data in the buffer and forward these data to other connected nodes when available.

Many application domains using DTNs have been studied since 2004, including outer-space communication, wildlife tracking, underwater monitoring, social networks, battlefields, developing regions, vehicular ad hoc networks, and pocket switched networks. 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 in India. The Haggle project [6] uses hand-held devices to exchange information among people. The CONDOR project [7] supported by the US Marine Corps studies communication in battlefields. The UMassDieselNet project [8] examines data transmission between buses moving around the campus of the University of Massachusetts.

DTN related standards are studied by the Delay Tolerant Networking Research Group (DTNRG) [9]. DTN architecture was introduced in RFC 4838 [10], with a new bundle layer added between the application layer and the transport layer. The bundle layer implements the store-and-forward mechanism. Data packets passing through bundle layer are grouped into basic units called bundles or messages. The protocol for this was specified in detail in RFC 5050 [11]. To support this bundle layer, the Licklider Transmission Protocol (or Long-haul Transmission Protocol, LTP) was introduced in RFCs 5325-5327 [12][13][14]. LTP provides retransmission-based reliable transmission over links having long message round-trip times (RTTs) and frequent interruptions.

A DTN topology can be represented by a graph $G = (V, E)$, where V denotes the set of network nodes and E denotes the set of edges or links. G is a time-varying graph which shows a snapshot of network connectivity at a certain point in time. An edge in G may stay connected for a period of time and then become disconnected. When an edge is connected, we say that the corresponding two nodes are *contacting* and the duration is called *contact duration*. The time duration between two successive contacts is called *inter-contact duration*.

Node contacts may follow a certain schedule: precise, approximate, implicit, or random [15]. The contact time in a satellite network is precise due to regular motion. A bus schedule is approximate due to occasional traffic jams or car accidents. Human contacts in a social network have some implicit rules. For example, two persons may come in contact frequently if they have many common friends. Node contacts under a random walk model are random.

The primary research issues in DTNs are briefly introduced below:

Routing: Data delivery only happens when two nodes are in contact in a DTN. This kind of network environment is also called an *opportunistic network* [16]. Knowledge about contact schedules becomes important to data routing. Many routing protocols for DTNs have been studied. In contrast with traditional routing approaches, DTN routing usually makes use of per-hop decisions and data replication.

Contact schedule: Knowing contact schedules facilitates routing decisions. Contact schedules may be unpredictable or predictable. An unpredictable schedule makes past contact history useless. A predictable schedule can be known through either oracle-based or history-based knowledge. Using oracle-based knowledge, future contacts at any time can be exactly predicted. Oracle-based routing approaches are discussed in [17]. In most DTN applications, contact schedules can only be approximately predicted according to past contact history. Much current research examines contact behavior in the

1530-2016/11 $26.00 © 2011 IEEE
DOI 10.1109/ICPPW.2011.19

human world. For example, Tournoux et al. [18] found the accordion phenomenon, in which the network topology expands and shrinks with time, from the trace file of rollerbladers. Chaintreau et al. [19] found that the distribution of inter-contact durations follows a power law with a heavy tail.

Buffer management: The store-and-forward mechanism used in a DTN requires buffer space at each node. Some routing approaches use extreme data replication and hence would quickly exhaust the buffer space. Dropping minor messages in a buffer is necessary when the buffer overflows. Moreover, two nodes may not have sufficient time to exchange all messages in their buffers within their contact duration. Transmitting major messages first is essential. How to determine the relative importance of each message in a buffer is a technical issue relating to buffer management.

In this paper, we study the existing routing strategies and buffer management policies for DTNs. Our work is different from other survey papers [15][20][21], in that we not only do a qualitative comparison but also a quantitative evaluation. A generic routing procedure is provided to explain different types of routing strategies. Moreover, we evaluate different buffering policies and recommend efficient ones.

The rest of this paper is organized as follows. In Section 2, we provide a classification of routing and buffering strategies. Section 3 discusses various mechanisms used in these strategies. Performance comparisons are presented in Section 4. Finally, we make some concluding remarks in Section 5.

II. Preliminary

DTN routing protocols have been widely discussed in recent years. Their routing operations can be divided into three main stages. When two nodes encounter one another, meta-data are exchanged. These meta-data include information about buffer management and routing decisions. 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 decision. Two major tasks are involved in buffer management: arrangement of message transmission order and message drop order. These two tasks can be accomplished together by sorting messages in a buffer using a certain strategy. The transmission order then starts from the head of the buffer. Four strategies can be used in message dropping:

- Drop front: drop the message at the head of the buffer.
- Drop end: drop the message at the end of the buffer.
- Drop tail: drop the new incoming message that is not yet entered into the buffer.
- Drop random: drop a message that is randomly selected from the buffer.

A variety of information items such as message size and message receipt time can be used individually or in combination as sorting indexes, as we will discuss further in Section 3.

A routing protocol is used to find a time-efficient delivery path from a source to a destination. In a DTN, the delivery time of a message over a link between two nodes is determined by four components [17]:

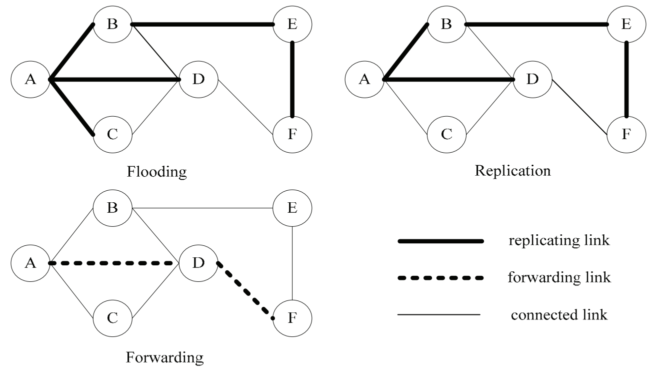


Figure 1. Three routing categories.

- *Connection waiting time*: time waiting for the link to be connected.
- *Buffer time*: time waiting for the message to be served in the buffer.
- *Transmission time*: time to deliver the message from the buffer to the link.
- *Propagation time*: time for the message to traverse the link.

To get a comprehensive view of various DTN routing strategies, we classify them according to four different dimensions: message copies, information type, decision type, and decision criterion.

Message copies: A routing scheme can be classified into three categories according to how many copies of the same message are created during the routing phase: *forwarding*, *replication*, and *flooding*. In a forwarding scheme, a single-copy message is forwarded through successive intermediate nodes to the destination. Only the source and the destination have this message copy. A replication scheme creates one copy of each message as it is forwarded to a next-hop node. Therefore, many intermediate nodes have a copy of the same message. A flooding scheme is an extreme replication scheme such that almost all nodes in the network have a copy of the same message. Examples are shown in Fig. 1, where nodes A and F are the source and the destination respectively.

Information type: To support routing decisions, each node in the network maintains either *local* or *global* routing information such as distance vectors or link states. One-hop and even two-hop nodes in the neighborhood need to exchange routing information in a local scheme; however, routing information is propagated to all nodes in a global scheme.

Decision type: A routing path can be decided on a *per-hop* or *source-node* basis. A source-node scheme works like a source routing protocol that selects a final routing path at the source node. In a per-hop scheme the routing path can be changed each time it is forwarded by an intermediate node.

Decision criterion: The decision of a routing path can be viewed as successive selections of next-hop nodes from a source toward a destination. For each decision, we need to evaluate whether a current encounter node is good enough to be a next-hop node. Different evaluation factors have been studied and can be classified into three categories: *node property*, *link property*, and *path property*.

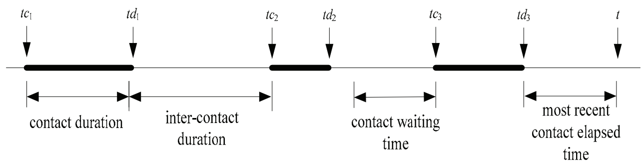


Figure 2. Contact related terms.

- *Node property*: number of recent contact nodes, betweenness, and buffer size.
- *Link property*: similarity, contact history and contact schedule, distance, relative moving direction, and relay willingness
- *Path property*: path delivery cost.

The number of recent contact nodes over an observation period indicates the contact rate and hence the degree of activity of a node in a social network. Betweenness [22] is an indicator of how important a node is in linking the other nodes in a network topology and is measured by the number of shortest paths passing through this node. Basically, a cut-point node [23] has a high betweenness value. The buffer size can be the remaining available size or the currently occupied size.

Similarity [22], contact history, and contact schedule are all used to predict how often and how long a link is established. If two nodes have several common neighbors in a network topology, the similarity between these two nodes is high. This implies the link between them is often established with high probability.

Let tc_i and td_i respectively denote the starting time and the ending time of the *i*th contact. Given recent k successive contact records for a pair of nodes with timing information observed within a time duration T: $\{(tc_1, td_1\}, (tc_2, td_2) ... (tc_k, td_k)\}$, the following contact related terms can be defined with their meanings shown in Fig. 2.

- *Average contact duration* (CD): average time duration of each contact, which is computed as,
$$\text{CD} = \frac{1}{k}\sum_{i=1}^{k}(td_i - tc_i)\,.$$
- *Average inter-contact duration* (ICD): average time duration between two successive contacts, which is computed as,
$$\text{ICD} = \frac{1}{k-1}\sum_{i=2}^{k}(tc_i - td_{i-1})$$
- *Average contact waiting time* (CWT): average time waiting for an expected contact from any point of time, which is computed as [24],
$$\text{CWT} = \frac{1}{2T}\sum_{i=2}^{k}(tc_i - td_{i-1})^2\,.$$
- *Contact frequency* (CF): number of contacts over the observation period T, which is computed as,
$$\text{CF} = k\,.$$
- *Most recent contact elapsed time* (CET): elapsed time from the moment when the most recent contact happened to the current point of time (denoted as t), which is computed as,
$$\text{CET} = t - td_k\,.$$

CD entails the link capacity of how many messages can be transferred during a contact opportunity. ICD, CWT, CF, and CET indicate the probability a link will be connected. CD, ICD, CWT, and CF can also be computed by exponential moving average over successive observation periods.

Distance is used to measure how close a node is currently to a destination. Relative moving direction determines whether two nodes move on parallel or perpendicular roads. Relay willingness [25] reflects how willing a node is to relay data for another node. The path delivery cost is based on the computation of a shortest path to a destination in hop count or delivery time. The node selection criterion then simply checks whether an encounter node is located on the shortest path.

III. ROUTING AND BUFFERING

A. Routing Strategies

In this sub-section, a general procedure that describes all types of routing strategies is provided. We then introduce specific DTN routing strategies.

1) Generic Routing Procedure

As mentioned in Section 2, DTN routing protocols can be classified into three categories according to the message copy dimension. However, all these protocols can be described using the same replication-based paradigm. In a replication scheme, each message generated from a source is associated with an initial quota. A node can conditionally duplicate a message with a non-zero quota in the buffer and forward the duplicated message to another encounter node. Moreover, the duplicated message is allocated part of the quota of the original message. Any message whose quota cannot be allocated any more is therefore transmitted to the destination by direct contact (or direct delivery [26]).

Let us introduce the following notations:

- v_i: node in the DTN network.
- m: message in the buffer.
- $Des(m)$: destination node of message m.
- $Src(m)$: source node of message m.
- QV_i^m : quota of m at v_i.
- Q_{ij}: quota allocation function from v_i to v_j.
- P_{ij}: conditional predicate of message copy or forward from v_i to v_j.

A replication routing scheme can be described in this way: As v_i contacts v_j and wishes to copy message m to v_j, v_i examines predicate P_{ij} to check whether v_j qualifies as a next-hop node. If P_{ij} is true and the condition $QV_j^m > 0$ is satisfied, m is copied from v_i to v_j. The related quota values are updated as follows:

$$QV_j^m = \lfloor Q_{ij} \times QV_i^m \rfloor,$$
$$QV_i^m = QV_i^m - QV_j^m\,.$$

If QV_i^m after the update is equal to zero, message m is removed from the buffer of v_i, which implies a forwarding action.

Fig. 3 gives an example, where P_{ij} and Q_{ij} have been assigned with each contact. Initially, A has a message with a quota of 2. When A contacts B, A forwards a message copy with a quota of 1 to B. When B contacts C, B does not copy the message to C because $QV_C^m = 0$. When B contacts D, B forwards a message copy with a quota of 1 to D. Then B removes its own message from the buffer because $QV_B^m = 0$.

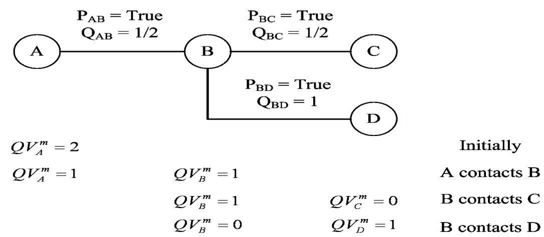


Figure 3. Quota allocation example.

TABLE I. QUOTA SETTINGS FOR DIFFERENT ROUTING SCHEMES.

Routing Strategy	Initial Quota	Quota Allocation Function
Flooding	∞	$Q_{ij} = \begin{cases} 1, \text{ if } p_{ij} = \text{true} \\ 0, \text{ if } p_{ij} = \text{false} \end{cases}$
Replication	k ($k > 0$)	$Q_{ij} = \begin{cases} \text{between 0 and 1, if } p_{ij} = \text{true} \\ 0, \text{ if } p_{ij} = \text{false} \end{cases}$
Forwarding	1	$Q_{ij} = \begin{cases} 1, \text{ if } p_{ij} = \text{true} \\ 0, \text{ if } p_{ij} = \text{false} \end{cases}$

By setting the initial quota and the quota allocation function as shown in Table 1, we can describe routing behaviors of flooding and forwarding in a similar way. In flooding, when a node meets a qualified node (i.e., P_{ij} is true), all messages in the buffer are copied to the encounter node. Moreover, the encounter node can continue copying these messages to other nodes and so on. Hence, we keep the quota value of a message as infinite conceptually and let the operations be true: $0\times\infty=0$ and $\infty-\infty=\infty$. As a result, a message will be flooded across the whole network if P_{ij} is always true. In forwarding, the initial quota of a message is one. Moreover, the quota allocation function always distributes the full quota, which causes a message to be dropped from a sending buffer.

Procedure contact (v_i, v_j) describes routing steps of v_i in encountering v_j. Indeed, v_j does the same thing as v_i at the same time and we ignore this part without loss of generality. Three meta-data items are exchanged in Step 1. Assume that each generated message has a unique global ID. The m-list summarizes the content of the buffer of one node by listing message IDs. By comparing two m-list items, we can avoid transmitting redundant messages between two nodes. In flooding and replication, 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 an i-list [27]. 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 to its i-list. Two contact nodes will exchange and merge their i-list records. The r-table contains routing information that needs to be propagated to other nodes.

After refreshing the buffer and the routing table in Steps 2 and 3, messages in the buffer are sorted in Step 4. Basically, those messages whose destinations are the node v_j have a high precedence to be transmitted first. In Step 5, each message in the buffer is examined with one of three consequences: ignored, copied, or forwarded.

Procedure contact (v_i, v_j)
Begin
1. Exchange meta-data: m-list, i-list, and r_table.
   m-list: summary vector of messages in the buffer.
   i-list: summary vector of messages that are known to reach their destinations.
   r_table: routing/contact information.
2. Update the routing table if necessary.
3. Remove messages that are indicated in the i-list of v_j from the buffer.
4. Order the messages in the buffer according to a certain sorting policy.
5. For each message m in the buffer from the head to the end,
   If m is in the m-list of v_j, ignore this message.
      If Des(m) == v_j
         Copy m to v_j.
         Remove m from the buffer.
      Compute $QV_j^m = \lfloor Q_{ij} \times QV_i^m \rfloor$.
      If Des(m) != v_j and $QV_j^m > 0$,
         Copy m to v_j.
         Update $QV_i^m = QV_i^m - QV_j^m$.
         Remove m from the buffer if QV_i^m is zero.

End.

2) Flooding Protocols

Flooding can be unconditional (P_{ij} is always true) or conditional. Unconditional flooding includes protocols Epidemic [28] and MaxProp [29]. Epidemic replicates all messages in the buffer that are not redundant to every contact node. If there are no buffer or bandwidth constraints in the network, this protocol performs optimally. However, limiting the buffer size at each node causes the dropping of some message copies and hence reduces the likelihood a message will reach its destination. Limiting the bandwidth increases message delivery time. MaxProp uses the same routing as Epidemic but improves buffer management. Messages in the buffer are sorted according to two indexes: hop count from the source to the current buffer node and delivery cost from the current buffer node to the destination. Messages with small hop counts are transmitted first, and messages with high delivery cost are dropped first. The computation of delivery cost necessitates the exchange of global routing information. The routing table at each node has size of $|E|$ entries at most.

Conditional flooding includes protocols PROPHET [30], Delegation [31], RAPID [32], BUBBLE Rap [33], DAER [34], and VR [35]. PROPHET measures the contact probability (between 0 and 1) of each link. The contact probability of a link increases when the link is up and decreases periodically when the link is down to reflect aging effect. Indirect contacts also contribute to some contact probabilities by using a transitive rule. The routing table at each node has size of $|V|-1$ entries at most. Let CP_i^m denote the contact probability between node v_i and the destination node of message m. The flooding predicate in PROPHET is represented as $P_{ij} = CP_i^m < CP_j^m$, which implies message replication to a node with a higher CP. This behavior is also called a *gradient routing*. The message replication will be blocked at a node with a locally maximal CP, and this is usually called a *local maximum problem*. In this case, the message can only reach the destination by direct contact with the destination. All gradient routing protocols suffer from this problem.

Delegation employs the use of CF which is the contact frequency of a node with the destination of a certain message. This routing copies a message to a node that has a higher CF than other nodes that have been contacted before. The flooding predicate is represented as $P_{ij} = \max[CF_i^m] < CF_j^m$, where $\max[CF_i^m]$ denotes the maximal CF value of message m in the contact history of v_i.

RAPID uses a utility function to evaluate the benefit of copying a message to a node. Different utility functions based on message size and delivery time are proposed for different optimization goals (e.g., minimizing end-to-end delay, minimizing the maximal end-to-end delay, etc). The flooding predicate is represented as P_{ij} = Does the utility value become better after a message is copied from v_i to v_j. The computation of a utility value is based on the estimation of the number of the copies of the same message in the network and the expected delivery time of these message copies to the destination. The computation cost of this is high and requires global exchange of many meta-data items.

BUBBLE Rap assigns each node a rank based on its betweenness and behaves like a gradient routing to copy messages to higher-rank nodes. The global ranking process entails significant cost.

DAER copies messages to all encounter nodes if the current message holding node is moving toward these message destinations and changes to forward mode otherwise. VR copies messages to those nodes driving on perpendicular roads with high probability. DAER and VR are only suitable for vehicular environments with the support of GPS (Global Positioning System).

3) Replication Protocols

Replication routing limits the maximum number of copies of the same message by setting an initial quota for each message. The setting of the quota is a tradeoff between resource consumption and message deliverability and hence is a challenge. If a message is never copied to the same node, the copy relationships among nodes in the network form a tree structure. The initial quota limits the tree size. Replication protocols include Spray&Wait [36], Spray&Focus [37], EBR [38], and SARP [39].

Both Spray&Wait and Spray&Focus use a binary quota allocation function (Q_{ij} =1/2). Messages with quota of one are forwarded to destinations through single hop and multiple hops in Spray&Wait and Spray&Focus, respectively. The link cost in evaluating a routing path is CET in Spray&Focus. EBR maintains an encounter value for each node, which is the average number of encounters with other nodes during an observation period. Q_{ij} is 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 encounter times between two nodes in a new way. A contact with a short duration and a long duration would respectively contribute zero and more than one to encounter times.

4) Forwarding Protocols

TABLE II. SUMMARY OF EXISTING DTN ROUTING PROTOCOLS.

Routing Protocol	Message Copies	Information Type	Decision Type	Decision Criterion
Epidemic	Flooding	None	Per-hop	None
MaxProp	Flooding	Global	Per-hop	Path
PROPHET	Flooding	Global	Per-hop	Link
BUBBLE Rap	Flooding	Global	Per-hop	Node
Delegation	Flooding	Local	Per-hop	Link
RAPID	Flooding	Global	Per-hop	Link
DAER	Flooding/ Forwarding	Local	Per-hop	Link
VR	Flooding	Local	Per-hop	Link
Spray& Wait	Replication/ Forwarding	None	Per-hop	None
Spray& Focus	Replication/ Forwarding	Local	Per-hop	Link
EBR	Replication	Local	Per-hop	Node
SARP	Replication/ Forwarding	Local	Per-hop	Link
SimBet	Forwarding	Local	Per-hop	Node/ Link
MED	Forwarding	Global	Source-node	Path
MEED	Forwarding	Global	Per-hop	Path
SSAR	Forwarding	Local	Per-hop	Link
FairRoute	Forwarding	Local	Per-hop	Node/ Link
PDR	Forwarding	Global	Source-node	Link
MFS,MRS, WSF	Forwarding	Local	Source-node	Node/ Link
Bayesian	Forwarding	Local	Per-hop	Link
SD-MPAR	Forwarding	Local	Per-hop	Link

In forwarding, a single-copy message is forwarded along a routing path to its destination. To derive the shortest path, each link (e_{ij}) in the network is measured with a cost (w_{ij}). Forwarding predicates have three general forms:

- Type 1: P_{ij} = Is e_{ij} on the shortest path from *Src*(m) to Des(m).
- Type 2: P_{ij} = Is e_{ij} on the shortest path from v_i to Des(m).
- Type 3: P_{ij} = Utility(v_i) < Utility(v_j).

Type 1 is for source-node forwarding. Types 2 and 3 are for per-hop forwarding. Types 1 and 2 involve the computation of routing path cost, so global routing information is exchanged. Type 3 mainly uses the node property or the local link property to evaluate a suitable next-hop node, so local information is enough. Also, type 3 behaves like gradient routing.

Example protocols of type 1 include PDR [40], MRS [41], MFS [41], WSF [41], and MED [17]. MED is an oracle-based routing that needs precise future information. MEED [24], a protocol of type 2, introduces *per-contact forwarding*, which is a variation of per-hop one. Per-contact forwarding forces forwarding decisions to be made after node encounters and temporally sets w_{ij} of the link between two current contact nodes to zero in computing a path cost. Different link cost models have been studied. For example, PDR uses the weighted average of CD and CWT. MRS uses CET, MFS uses the inverse of CF, and WSF uses the ratio of the remaining buffer size to CF. MEED uses CWT.

Protocols of type 3 include SimBet [22], SSAR [25], FairRoute [42], Bayesian [43], and SD-MPAR [44]. Different kinds of information are used in computing a utility value.

SimBet combines betweenness and similarity. SSAR uses relay willingness and ICD. FairRoute uses interaction strength and queue size. The interaction strength indicates the likelihood a contact will be sustained over time. Bayesian is based on historical successful relay counts. SD-MPAR combines the distance and moving direction relative to the destination.

Based on our classification dimensions, we investigate all above existing DTN routing protocols and delineate their features in Table 2.

B. *Buffer Management*

Determination of transmission and drop order is a major task in buffer management, and it can be done based on a sorting policy. Below, we list some sorting indexes by which messages in the buffer can arranged in **ascending order**.

- *Received time:* time for a message to be received by the current buffer node. The FIFO (First-In-First-Out) policy is modeled by this index.
- *Hop count*: number of hops for a message to travel from the source to the current buffer node.
- *Remaining time*: amount of time remaining until the death of a message.
- *Number of copies*: number of copies of a message in the network.
- *Delivery cost*: cost for a message to be delivered from the current buffer node to the destination.
- *Message size*: size of a message.
- *Service count*: number of transmission events for a message in the buffer. This index is used for service fairness by simulating round-robin behavior.
- *Distance to the destination*: distance between the current buffer node and the destination node of a message.

An evaluation has been done [45] of some of the above indexes as a drop order policy. The weakness of this work is that their evaluations were not performed under a real contact environment. An optimal drop policy is studied in [46], but their method requires information about the total number of nodes and the exact number of message copies for each message delivered in the network. Moreover, the average contact rate in the network is assumed to be exponentially distributed. Although heuristic solutions are mentioned to approximate these numbers, large amounts of meta-data are exchanged and recorded. RAPID, which studies the transmission order, has the same problem. MaxProp is the only paper that considers two sorting indexes together related to buffer management. The buffer space is adaptively divided into two parts which are separately sorted by hop count and delivery cost; as a result, the operations are a little bit complicated.

In this paper, we examine each of these indexes (except for the distance factor, which requires additional location information) by using real contact traces. The delivery cost is in the format of the inverse of contact probability used in PROPHET. We use a utility function as below to evaluate the importance of message m in the buffer:

$$Utility(m) = \frac{1}{\text{Index 1+Index 2+...}}$$

Messages with higher utility values are transmitted first and messages with lower utility values are dropped first.

In a fully distributed environment, getting the number of copies of each delivered message is difficult. Here, we introduce a simple method called *MaxCopy*, which has a low storage-space requirement, to estimate this value. Each message in the buffer is associated with a copy counter. For instance, node A generates a new message m and initiates the copy counter at 1. When message m is copied from node A to node B, both nodes A and B update their copy counters to 2. After node A copies m to node C, the copy counters in nodes A and C are updated to 3. When nodes B and C are contacting, they find their copy counters of message m are different and then update to the maximal value of 3.

IV. PERFORMANCE EVALUATION

To study the performances of routing and buffering schemes, we choose to use the Opportunistic Network Environment (ONE) simulator [47]. ONE is JAVA-based open-source software, which has provided implementation codes of some routing protocols. Two real contact trace files are downloaded from the CRAWDAD webpage [48] and are denoted as Infocom and Cambridge in the paper. These two traces record contact activities of 3~4 days at the conference venue of 2005 IEEE INFOCOM and at the computer lab of University of Cambridge. Infocom involves 268 nodes (internal + external) and Cambridge involves 223 nodes (internal + external).

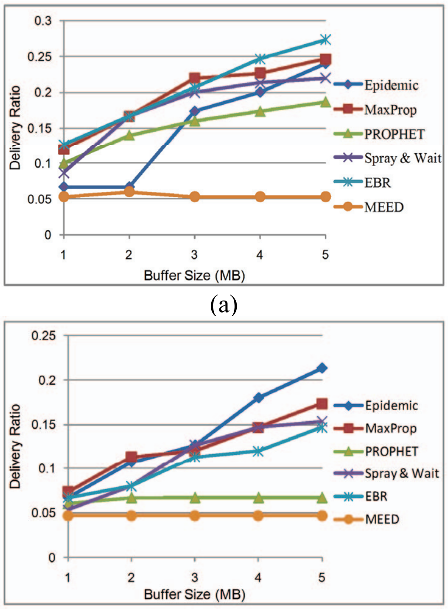
Additionally, we generate node contact events in a vehicular environment (denoted as VANET in the experiments). Under a street model, 100 vehicles are generated and their moving patterns are controlled by the tool VanetMobiSim [49]. The average moving speed of a vehicle is 60 km/hr. Two nodes are in contact if the distance between them is shorter than 200 m (i.e., the wireless transmission radius).

In our experiments, 150 messages of size 50 kB to 500 kB each are generated at a time interval of 30 s after a system warm-up time. Sources and destinations of these messages are randomly selected from the network nodes. The transmission rate of each link is 250 kBps.

Three cost metrics are used in our performance evaluation. Note that the received message below indicates the first message copy arrived at the destination.

- *Delivery ratio*: ratio of the total number of messages received at destinations to the total number of messages sent from sources.
- *Delivery throughput*: average data delivery rate in byte per second for those messages successfully received at the destinations.
- *End-to-end delay*: average delivery time for a message from source to destination.

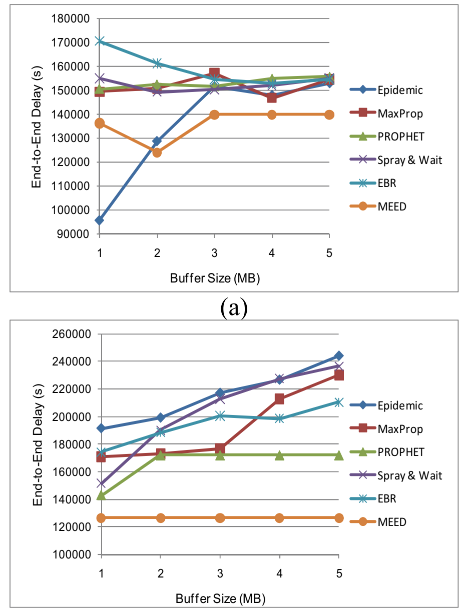
First, we evaluated several routing protocols selected from each category: Flooding (Epidemic, MaxProp, and PROPHET), Replication (Spray&Wait and EBR), and Forwarding (MEED). For a fair comparison, these routing protocols are all implemented with the i-list mechanism. The sorting index in the buffer was based on the message received time and the drop policy was Drop Front. To save space, only the results of delivery ratio and end-to-end delay are presented.



(a)

(b)

Figure 4. Delivery ratio: (a) Infocom (b) Cambridge.



(a)

(b)

Figure 5. End-to-end delay: (a) Infocom (b) Cambridge.

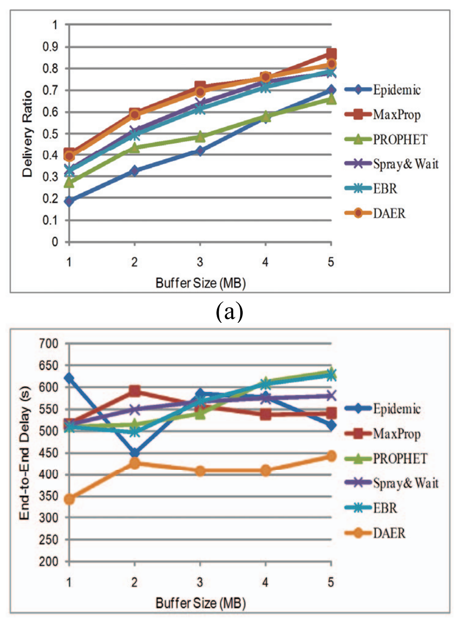
Performance trends are similar in the results of delivery ratio and delivery throughput.

Figs. 4a and 4b show that MaxProp and EBR have good performance in Infocom, while Epidemic and MaxProp have good performance in Cambridge. Infocom represents frequent contact events, so replication routing is suitable. Cambridge represents rare contact events, so flooding routing is suitable. MaxProp with suitable buffer management performs well in both traces. MEED performs worst, because many messages could not reach their destinations due to long delivery paths. Epidemic had poor performance when the buffer size was small, because a large number of message copies overwhelms small buffers.

By analyzing these real contact traces, we observed several phenomena. Not all nodes were in contact directly or indirectly, so many messages could not reach their destinations. Some pairs of nodes were in frequent contact with each other at the beginning and stopped any contacts after a certain period. Some contacts had a very long inter-contact duration. These phenomena resulted in inaccurate predictions of contact behaviors.

For example, MaxProp lacks an aging function to degrade previously accumulated contact frequencies. The delivery cost used in buffer management has an estimation error, which causes an incorrect drop order. PROPHET uses an aging function that periodically degrades the contact frequency. An occasional long inter-contact period will fully erase previous values. Most routing strategies based on contact histories cannot adapt to sudden irregular behaviors.

Figs. 5a and 5b show the results of end-to-end delay. MEED has a low delay value, because there were some messages with short delivery paths in the network. Other messages with long delivery paths could not reach their destinations. Replication schemes have a higher delay than flooding ones in certain cases, because the last hop to the destination is by direct contact and waiting for this contact might be time consuming.



(a)

(b)

Figure 6. VANET: (a) Delivery ratio (b) End-to-end delay.

In Fig. 6, we show performance comparisons in our simulated vehicular environment. MEED is replaced by DAER which is a location-based routing scheme. DAER has the same good performance as MaxProp in delivery ratio and more importantly DAER has the less delay. The selection of one relay node that is closer to the destination in DAER benefits the reducing of end-to-end delay.

TABLE III. DIFFERENT BUFFERING POLICIES.

Policy	Sorting Index	Transmission Order	Drop Order
Random_DropFront	Received time	Transmit random	Drop front
FIFO_DropTail	Received time	Transmit front	Drop tail
MaxProp	Hop count and delivery cost	Transmit front	Drop end
UtilityBased	Utility value	Transmit front	Drop end

Next, we compare different buffering policies as indicated in Table 3, using the same routing of Epidemic. Epidemic is a flooding-based routing scheme and can show the significant effect of buffering management. Transmit front in the table indicates the transmission order is from the head of the buffer. Transmit random indicates that a randomly selected message in the buffer is transmitted each time.

The UtilityBased policy is our recommended one, and the utility functions used in our experiments are given below:

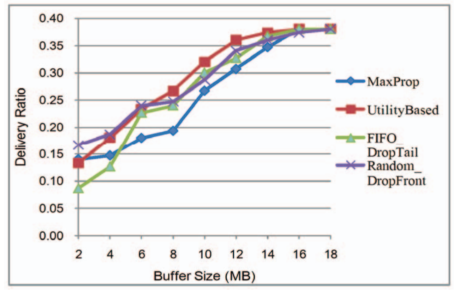
$$Utility_{delivery\_ratio}(m) = \frac{1}{\text{Message size} + \text{Number of copies}}$$

$$Utility_{throughput}(m) = \frac{1}{\text{Number of copies}}$$

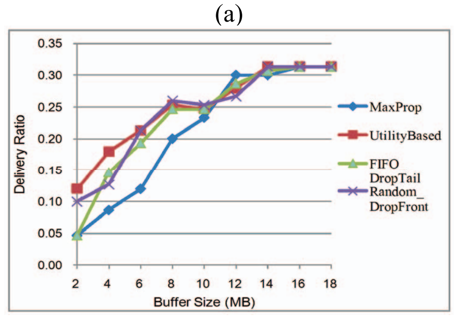
$$Utility_{delay}(m) = \frac{1}{\text{Delivery cost}}$$

The derivation of these utility functions is based on our pre-test on different combinations of sorting indexes. As can be seen, different utility functions are used for different cost metrics. For delivery ratio, two sorting indexes were used: message size and number of copies. The index of message size is consistent with the shortest-job-first concept. The current number of copies of a message reflects the progress of message spreading. A small copy number and a large copy number respectively indicate an early stage and a late stage of message spreading. This benefits performance by enforcing a scheduling rule of encouraging early-stage messages and suppressing late-stage ones. For delivery throughput, the message size becomes the least important, since small messages contribute less to data rate. The number of copies is still a significant index. For end-to-end delay, the delivery cost is the only index that directly reflects the path delivery time of a message, and hence is the most significant.

Comparison results are shown in Figs. 7~9. Apart from the UtilityBased policy, Random_DropFront has good performance on delivery ratio and delivery throughput in both Infocom and Cambridge. MaxProp, which involves path delivery cost, has good performance in end-to-end delay. UtilityBased performs best in Infocom and is close to the best in Cambridge on average for each cost metric. We get similar results, which are not shown here due to space limit, by changing the routing strategy to Spray&Wait. If a forwarding scheme, MEED, is used, all policies perform similarly due to the lower requirement for buffer space.

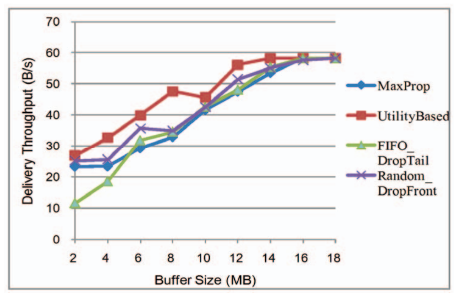


(a)

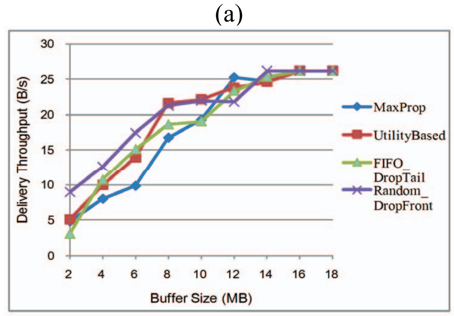


(b)

Figure 7. Delivery ratio: (a) Infocom (b) Cambridge.

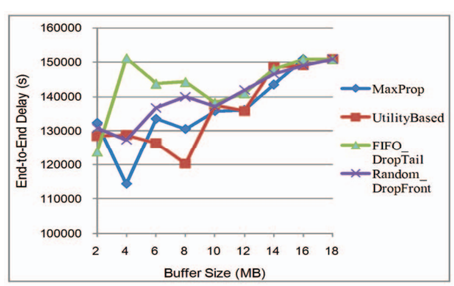


(a)

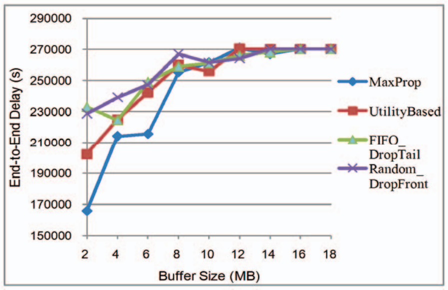


(b)

Figure 8. Delivery throughput: (a) Infocom (b) Cambridge.



(a)



(b)

Figure 9. End-to-end delay: (a) Infocom (b) Cambridge.

V. CONCLUDING REMARKS

In challenged networks where communications are subject to delays and disruptions, end-to-end routing operations become difficult. DTN related techniques have been studied to compensate for the traditional network architecture which causes malfunctions due to intermittent connections. In this paper we have identified three main DTN routing strategies: flooding, replication, and forwarding. A generic routing procedure is provided that can describe various DTN routing schemes based on a replication-based paradigm. We propose a clear classification of existing routing schemes. Many routing schemes use knowledge derived from contact histories such as contact frequency and contact waiting time in routing decisions.

We evaluated several routing strategies using real contact traces and demonstrated the following: Flooding and replication are better than forwarding. Irregular contact behaviors will largely degrade the performance of routing strategies using contact histories.

In buffer management, we used a utility-based sorting policy and got the following results: The delivery cost is an important sorting index that dominates end-to-end delay. The number of message copies greatly influences delivery throughput. Both the number of message copies and message size have a great influence on delivery ratio.

In the following, we provide some suggestions on the routing and buffering design:

- Single contact vs. multiple contacts: Most DTN routing strategies make routing decisions for individual contact node without considering other surrounding contact nodes. The decision question is limited to the type: "Do I forward/copy messages to the current contact node?" In certain network types, for example, wireless networks, a node is easy to have more than one neighbor. We highlight the point that more complex decision questions would improve routing performance. Example questions include "Which of my current neighbors are good enough to be next-hop nodes?" and "How does a quota value be allocated to multiple next-hop nodes?"
- Network-dependent strategies: We claim that routing strategies should be network-dependent. A contact-based strategy taking advantage of contact histories is suitable for social networks with regular or implicit behaviors. A motion-based strategy that considers location and direction information is suitable for vehicular networks or mobile ad hoc networks. In some DTNs, there exist separated stationary nodes and a few of mobile nodes. These mobile nodes act as message ferries to transport messages among stationary nodes. The routing strategy would rely on the moving schedules of these mobile nodes.
- Single connection vs. multiple connections: The transmission order of messages in the buffer is mostly determined for a single connection. If multiple concurrent connections are available, fairness and priority issues crossing different connections become potential.

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] D. Mohney, "Soaring Condor Relieves Headaches," *Mobile Radio Technology*, Jun. 2004.

[8] UMass DieselNet, http://prisms.cs.umass.edu/dome/umassdieselnet/

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

[10] 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.

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

[12] S. Burleigh, M. Ramadas, and S. Farrell, "Licklider Transmission Protocol - Motivation," IETF RFC 5325, Sept. 2008.

[13] M. Ramadas, S. Burleigh, and S. Farrell, "Licklider Transmission Protocol - Specification," IETF RFC 5326, Sept. 2008.

[14] S. Farrell, M. Ramadas, and S. Burleigh, "Licklider Transmission Protocol - Security Extensions," IETF RFC 5327, Sept. 2008.

[15] E. P. C. Jones and P. A. S. Ward, "Routing Strategies for Delay-Tolerant Networks," submitted to Computer Communication Review, http://www.ccng.uwaterloo.ca/~pasward/Publications/dtn-routing-survey.pdf.

[16] C. M. Huang, K. C. Lan, and C. Z. Tsai, "A Survey of Opportunistic Networks," *in Proc. IEEE AINA Workshop*, pp. 1672-1677, Mar. 2008.

[17] S. Jain, K. Fall, and R. Patra, "Routing in a Delay Tolerant Network," *in Proc. ACM SIGCOMM Conf.*, pp. 145-158, Aug. 2004.

[18] P.-U. Tournoux, J. Leguay, F. Benbadis, V. Conan, M.D. de Amorim, and J. Whitbeck, "The Accordion Phenomenon: Analysis, Characterization, and Impact on DTN Routing," *in Proc. IEEE INFOCOM Conf.*, pp. 1116-1124, Apr. 2009.

[19] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of Human Mobility on the Design of Opportunistic Forwarding Algorithms," *in Proc. IEEE INFOCOM Conf.*, pp. 1-13, Apr. 2006.

[20] J. Shen, S. Moh, and I. Chung, "Routing Protocols in Delay Tolerant Networks: A Comparative Survey," *in Proc. Intl. Conf. Circuits/Systems, Computers and Communications*, pp. 1577-1580, Jul. 2008.

[21] R. J. D'Souza and J. Jose, "Routing Approaches in Delay Tolerant Networks: A Survey," *Intl. Journal of Computer Applications*, vol. 1, no. 17, pp. 8-14, 2010.

[22] 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.

[23] L. Ding, B. Gu, X. Hong, and B. Dixon, "Articulation Node Based Routing in Delay Tolerant Networks," *in Proc. IEEE PerCom Conf.*, pp. 1-6, Mar. 2009.

[24] 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.

[25] Q. Li, S. Zhu, and G. Cao, "Routing in Socially Selfish Delay Tolerant Networks," *in Proc. IEEE INFOCOM Conf.*, pp. 1-9, Mar. 2010.

[26] T. Spyropoulos, K. Psounis, and C.S. Raghavendra, "Single-Copy Routing in Intermittently Connected Mobile Networks," *in Proc. IEEE Intl. Conf. Sensor and Ad Hoc Communications and Networks*, pp. 235-244, Oct. 2004.

[27] 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.

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

[29] 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.

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

[31] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot, "Delegation Forwarding," *in Proc. ACM MobiHoc Conf.*, pp. 251-260, May 2008.

[32] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "Replication Routing in DTNs: A Resource Allocation Approach," *IEEE/ACM Trans. Networking*, vol. 18, no. 2, pp. 596-609, Apr. 2010.

[33] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE Rap: Social-Based Forwarding in Delay Tolerant Networks," *in Proc. ACM MobiHoc Conf.*, pp. 241-250, May 2008.

[34] H. Y. Huang, P. E. Luo, M. Li, D. Li, X. Li, W. Shu, and M. Y. Wu, "Performance Evaluation of SUVnet With Real-Time Traffic Data," *IEEE Trans. Vehicular Technology*, vol. 56, no. 6, pp. 3381-3396, Nov. 2007.

[35] H. Kang and D. Kim, "Vector Routing for Delay Tolerant Networks," *in Proc. IEEE VTC Conf.*, pp. 1-5, Sept. 2008.

[36] 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.

[37] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility," *in Proc. IEEE Pervasive Computing and Communications Workshops*, pp. 79-85, Mar. 2007.

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

[39] 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.

[40] L. Yin, H. M. Lu, and Y. D. Cao, "Probabilistic Delay Routing for Delay Tolerant Networks," *in Proc. Intl. Conf. Advanced Communication Technology*, pp. 191-195, Feb. 2008.

[41] D. Henriksson, T. F. Abdelzaher, and R. K. Ganti, "A Caching-Based Approach to Routing in Delay-Tolerant Networks," *in Proc. Intl. Conf. Computer Communications and Networks*, pp. 69-74, Aug. 2007.

[42] J. M. Pujol, A. L.Toledo, and P. Rodriguez, "Fair Routing in Delay Tolerant Networks," *in Proc. IEEE INFOCOM Conf.*, pp. 837-845, Apr. 2009.

[43] S. Ahmed and S. S. Kanhere, "A Bayesian Routing Framework for Delay Tolerant Networks," *in Proc. IEEE WCNC Conf.*, pp. 1-6, Apr. 2010.

[44] L. Yin, Y. Cao, and W. He, "Similarity Degree-Based Mobility Pattern Aware Routing in DTNs," *in Proc. Intl. Conf. Intelligent Ubiquitous Computing and Education*, pp. 345-348, May 2009.

[45] A. Lindgren and K. S. Phanse, "Evaluation of Queueing Policies and Forwarding Strategies for Routing in Intermittently Connected Networks," *in Proc. Intl. Conf. Communication System Software and Middleware*, pp. 1-10, 2006.

[46] A. Krifa, C. Baraka, and T. Spyropoulos, "Optimal Buffer Management Policies for Delay Tolerant Networks," *in Proc. IEEE SECON Conf.*, pp. 260-268, Jun. 2008.

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

[48] The Community Resource for Archiving Wireless Data At Dartmouth (CRAWDAD), http://uk.crawdad.org/index.php

[49] VanetMobiSim, http://vanet.eurecom.fr/