Efficient Routing in Delay Tolerant Networks with Correlated Node Mobility

Eyuphan Bulut, Sahin Cem Geyik, Boleslaw K. Szymanski
Department of Computer Science and Center for Pervasive Computing and Networking
Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY 12180, USA
{bulute, geyiks, szymansk}@cs.rpi.edu

***Abstract*—In a delay tolerant network (DTN), nodes are connected intermittently and the future node connections are mostly unknown. Since in these networks, a fully connected path from source to destination is unlikely to exist, message delivery relies on opportunistic routing. However, effective forwarding based on a limited knowledge of contact behavior of nodes is challenging. Most of the previous studies looked at only the pairwise node relations to decide routing. In contrast, in this paper, we analyze the correlation between the meetings of each node with other nodes and focus on the utilization of this correlation for efficient routing of messages. We introduce a new metric called *conditional intermeeting time*, which computes the average intermeeting time between two nodes relative to a meeting with a third node using only the local knowledge of the past contacts. Then, we show how we can utilize the proposed metric on the existing DTN routing protocols to improve their performance. For shortest-path based routing protocols in DTNs, we propose to route messages over conditional shortest paths in which the link cost between nodes are defined by conditional intermeeting times. Moreover, for metric-based forwarding protocols, we propose to use conditional intermeeting time as an additional delivery metric while making forwarding decisions of messages. Our trace-driven simulations on three different datasets show that the modified algorithms perform better than the original ones.**

I. Introduction

Delay tolerant networks (DTN) [1]-[3], also called as intermittently connected mobile networks (ICMN), are wireless networks in which at any given time instance, the probability that there is an end-to-end path from a source to a destination is low. This is caused by the high mobility and low density of the nodes in the network. This property makes the routing of messages in DTNs a challenging problem. The conventional solutions do not generally work in DTNs because they assume that the network is stable most of the time and failures of links between nodes are infrequent. Therefore, the routing problem is still an active research area in DTNs [4].

With the advent of DTNs, *store-carry-and-forward* paradigm has been widely applied to routing. When a node has a message but no connection to any other node in the network, it stores the message in its buffer and carries the message until it meets a new node that does not have this message. If the encountered node is assessed to be useful in the delivery of the message, then the message is transferred to it. Several DTN routing algorithms based on this paradigm have been proposed recently. In each, different techniques (single-copy [5], multi-copy [6]-[8], erasure coding [9] [10]) with slightly different goals have been utilized. A survey of previous DTN routing algorithms can be found in [11].

Since DTNs consist of mobile agents that contact intermittently, recent studies on DTN routing have focused on the analysis of real mobility traces (human [16] [17], vehicular [18] etc.) and utilized extracted characteristics of the mobile objects on the design of forwarding algorithms for DTNs. These designs proposed in previous work led us to the following conclusions:

- The pairwise intermeeting times between the nodes usually follow a log-normal distribution [19] [20] so that they are not memoryless[1]. This makes future contacts of nodes dependent on their past.
- Most of the previous routing protocols that make the forwarding decisions at node meetings use a metric (encounter frequency [15], time elapsed since last encounter [21] [22], social similarity [23]-[25] etc.) computed from the pairwise relations of nodes with the destination node only. They do not use the information of 'meeting with each other' while computing their delivery metrics.
- The mobility of many real objects are non-deterministic but periodic [27]. For example, in a cyclic MobiSpace [27], if two nodes were frequently in contact at a particular time in previous cycles, then they are likely to meet around the same time in the next cycle.

The above three key observations motivated us to do the research presented in this paper that utilizes the correlation between the meetings of nodes for designing more efficient routing algorithms. Hence, we introduce a new link metric, *conditional intermeeting time*, that computes the average intermeeting time between two nodes relative to a meeting

This research was sponsored by US Army Research Laboratory and the UK Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001 and under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors, and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the U.S. Government, the UK Ministry of Defense, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

[1] Assume that X is the random variable representing the intermeeting time between two nodes, then the meetings are memoryless if $P(X > s+t \mid X > t) = P(X > s)$ for $s, t > 0$ holds.

with a third node using only the local knowledge of the past contacts. For example, with $\tau_A(D, B)$, we define the conditional intermeeting time of node A with destination D given the condition that it has just B as the average time it took in the past (computed from previous contact history) to meet with D after its meeting with B. Hence, at each meeting of node A with a potential destination node D, we compute the meeting frequency of A with D conditional on meetings with all the other nodes node A met since its last meeting with node D. In the paper, we discuss how this metric addresses all the above mentioned three key observations.

Moreover, note that this definition of intermeeting times is also meaningful in terms of the context of routing of messages because the messages are received from a node and given to another node on the way towards the destination. Moreover, the conditional intermeeting time represents the duration that a node holds the message throughout routing.

We provide the analysis of conditional intermeeting time and discuss when and why it can be beneficial for providing better information to nodes within the context of routing. We also present statistical results from three different data sets (RollerNet [20], Cambridge [30], Haggle [32]) which contain the contact traces of real objects logged during real life experiments.

To show the benefit of this metric, we propose improvements on existing DTN routing protocols. Firstly, for the algorithms which route messages over shortest paths (SP) [12] [14], we propose to use conditional intermeeting times rather than standard intermeeting times and route the messages over conditional shortest paths (CSP). Secondly, for the algorithms which make message forwarding decisions depending on a delivery metric (we call them metric-based forwarding algorithms), we propose to use conditional intermeeting time as a secondary delivery metric and allow the forwarding of messages if and only if both the algorithm's original delivery metric and conditional intermeeting time suggest to forward the message to the encountered node.

Through real-trace-driven simulations on three data sets, we compare the modified protocols with the original ones. The results show that modifications improve performances of those protocols. This shows that the conditional intermeeting time represents inter-node link costs well (in the context of routing) guiding forwarding decisions effectively while routing a message.

In short, the contributions of this paper are four-fold:

1) A new metric, conditional intermeeting time is proposed; it computes the average intermeeting time between two nodes relative to a meeting with a third node using only the local knowledge of the past contacts.
2) The analysis of the proposed metric is presented to show significance of the metric for routing.
3) Modification of the existing DTN routing protocols by using conditional intermeeting time metric is discussed.
4) Real-trace-driven simulations are performed and the performance improvement achieved in the updated versions of the existing algorithms on three different data sets are shown. Thus, the benefit of conditional intermeeting time in making effective decisions in DTN routing is shown experimentally.

The remaining of the paper is organized as follows. In Section II, we describe the proposed metric in detail and explain how a node can compute it from its previous contacts. In Section III, we give the analysis of proposed metric. In Section IV, we describe how the conditional intermeeting time can be used to modify existing DTN routing algorithms for performance improvement. In Section V, we present the details of real-trace-driven simulations in which we compare the performance of modified algorithms with the performances of original algorithms. Finally, we offer conclusion and outline the future work in Section VI.

II. Conditional Intermeeting Time

Recently, the research community working on routing algorithms in DTNs has focused on the analysis of real mobility traces to understand the main characteristics of mobile objects. Many experiments in different environments (office [19], conference [32], city [30], skating tour [20]) with different objects (human [16], bus [18], zebra [31]) and with variable number of attendants were performed. From the analysis of the data sets collected during these experiments, significant results about the aggregate and pairwise mobility characteristics of real objects were obtained.

As opposed to the random mobility models [26] according to which the intermeeting time between pairs follow an exponential distribution, recent analysis [19] [20] on real mobility traces showed that the intermeeting times between most (more than 99% in many datasets) of the node pairs fit to log-normal distribution. However, this revokes the validity of a common assumption [8] [13] [22] that the pairwise intermeeting times are exponentially distributed and memoryless, and it makes the pairwise contacts between nodes dependent on their pasts.

As a first consequence of this, the residual time until the next meeting of two nodes can be predicted more accurately depending on the condition that they have not met during the last t time units [19]. However, the dependence of future contacts on the past of nodes can provide more than this. The nodes can also benefit from their past contacts to predict their future contacts more accurately.

Previous work have used pairwise node relations in the design of DTN routing protocols while making effective forwarding decisions. The more frequently a node meets with destination, the higher probability it can deliver a message to the destination. Therefore, in the meeting of two nodes, keeping the message in the node with smaller average intermeeting time with destination increases the delivery probability. However, an important aspect here is that nodes receive messages from a node and forward it to another node. That is, the duration that a node holds a message is between its contacts with two different nodes.

Depending on the above facts, we propose a new metric called *conditional intermeeting time* to define the node relations more precisely within the context of routing. This metric

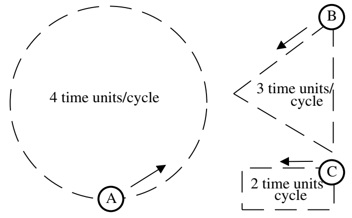


Fig. 1: A physical cyclic MobiSpace with a common motion cycle of 12 time units.

computes the average intermeeting time between two nodes relative to a meeting with a third node using only the local knowledge of the past contacts[2].

The proposed metric can also provide more accuracy if the nodes move in a cyclic MobiSpace [27]. According to the definition of cyclic MobiSpace, if two nodes contact frequently at a particular time in previous cycles, the probability that they will be in contact around the same time in the next cycle is high. In Figure 1, a sample cyclic MobiSpace with three objects is illustrated. The common motion cycle is 12 time units. In this example, the discrete probabilistic contacts between A and B happen in every 12 time units (1, 13, 25..) and the discrete probabilistic contacts between B and C occur in every 6 time units (2, 8, 14..). When we consider intermeeting time between nodes B and C, we can expect that node B will forward its message to node C in 6 time units, however conditional intermeeting time of B with C based on the condition that it has met (received the message from) node A lets us know that the message will be forwarded to node C within 1 time unit.

In a DTN, each node can keep and update the list of its contacts and the previous meeting times with each of them during its lifetime. Then, utilizing this information, each node can find the values of both of their standard and conditional intermeeting times with their neighbors.

Let $\tau_A(B|C)$ denote the conditional intermeeting time of node A with node B given that it has met node C. It represents the average time that elapses from the time A has met C to the time it first meets B. Hence, the standard intermeeting time between A and B can be denoted as $\tau_A(B|B)$ (or shortly $\tau_A(B)$). To find the conditional intermeeting time of $\tau_A(B|C)$, each time node A meets node C, it starts a different timer. When it meets node B, it takes the sum of the values on these timers and also notes the number of timers it has started before deleting them. This computation is repeated again each time node C is encountered. Then, dividing the sum of time values collected from each timer by the total count of all timers used gives the value of $\tau_A(B|C)$. We can also use a sliding window with an appropriate size over the past contacts [14] to make the conditional intermeeting value more local and current.

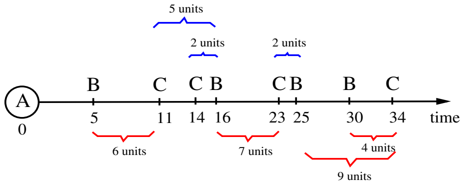


Fig. 2: Example meeting times of node A with nodes B and C. While the values in the upper part are used in the computation of $\tau_A(B|C)$, the values in the lower part are used in the computation of $\tau_A(C|B)$.

Consider Figure 2 which shows the sample contact times of a node A with its neighbors B and C. Node A meets with node B at times $\{5, 16, 25, 30\}$ and with node C at times $\{11, 14, 23, 34\}$. Then, following the procedure described above, $\tau_A(B|C) = 3$ and $\tau_A(C|B) = 6.5$ units.

III. Analysis

In this section, we give the analysis of conditional intermeeting time and show why it is significant in accurate prediction of future meetings within the context of routing.

Assume that node A has two different contacts, B and C, and the sets S_B and S_C include the meeting times of node A with nodes B and C in order during the time frame [0, T], respectively.

$$S_B = \{B_1, B_2, B_3, \ldots B_n\}, \text{n meetings}$$
$$S_C = \{C_1, C_2, C_3, \ldots C_m\}, \text{m meetings}$$

Furthermore, assume that the intermeeting time of node A with node B is well represented by a random variable X_1 while the intermeeting time of node A with C is well represented by a random variable X_2 with the CDF $D_2(x)$ and probability density $d_2(x) = D_2'(x)$.

To find $\tau_A(C|B)$, we need to compute the following (without loss of generality, we assume that $C_m \geq B_n$):

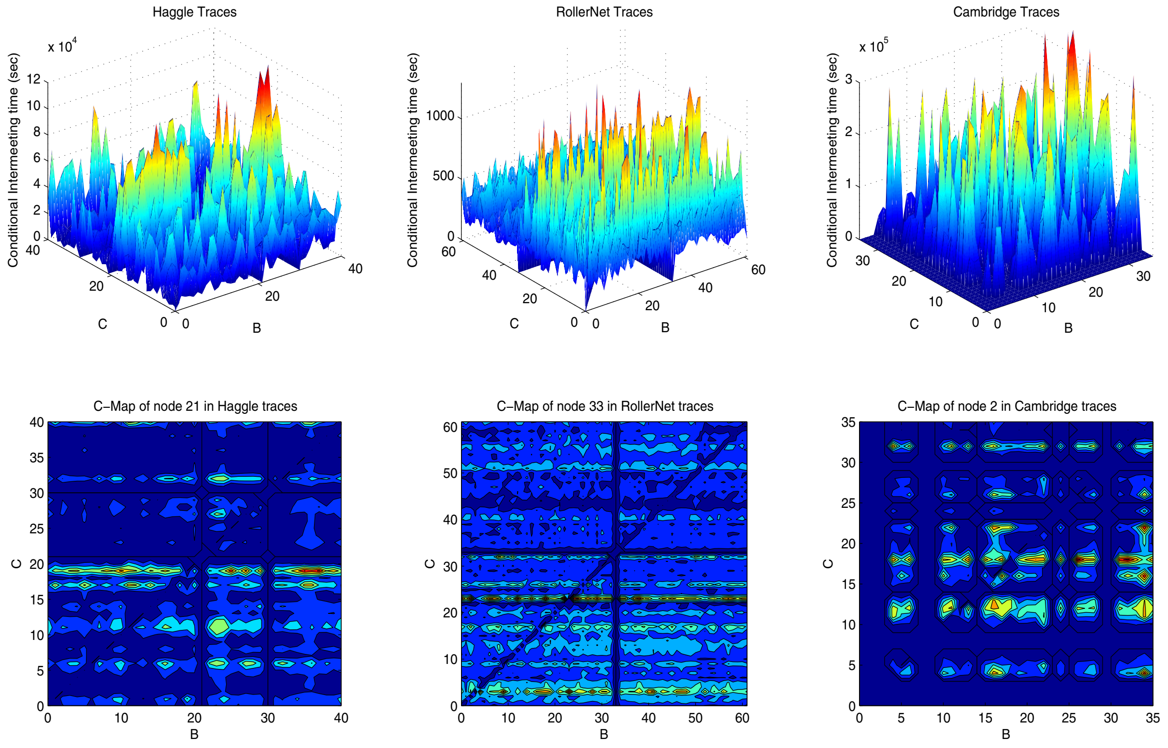
$$\tau_A(C|B) = \frac{\sum_{k=1}^{m}(C_{d(k)} - B_k)}{m},$$

where, $d(k) = \min\{i : C_i \geq B_k\}$.

The conditional intermeeting time of node A with node C under the condition that it has met node B is then a random variable that we will denote as Y. Then $\tau_A(C|B) = E[Y]$. Let's consider j^{th} meeting of node A with node B and denote $t = B_j - C_{d(j)-1}$. Consider a family of random variables $Y(t)$'s with CDFs $D_t(x)$ and probability densities $d_t(x)$, then:

$$D_t(y) = \frac{D_2(y+t) - D_2(t)}{1 - D_2(t)}$$
$$d_t(y) = \frac{d_2(y+t)}{1 - D_2(t)}$$

[2] Note that, the proposed metric is different than *conditional residual time* (CRT) proposed by Srinivasa et al. in [19]. While CRT computes the remaining time for the meetings of two nodes based on the condition that the nodes have not met each other for some time, our metric computes the residual time to the next meeting of two nodes based on the condition that one of the nodes met with a third node. As we will see in analysis section, CRT is a special case of conditional intermeeting time in which the distribution of passed time since the last meeting is not considered.



(a) C-Map of node 21 in Haggle traces (b) C-Map of node 33 in RollerNet traces (c) C-Map of node 2 in Cambridge traces

Fig. 3: C-Maps of popular nodes in three datasets. In figures, B represents the id of the node already met and C represents the id of the node to be met.

$$E[Y(t)] = \frac{\int_{y=0}^{\infty} y d_2(y+t)dy}{1-D_2(t)}$$

$$= \frac{\int_{y=t}^{\infty}(1-D_2(y))dy}{1-D_2(t)}$$

When X_1 and X_2 are exponentially distributed random variables, then $D_t(x) = D_2(x)$, showing the memoryless property of exponential distribution. However, as the previous work suggests, X_1 and X_2 fit well with log-normal distribution. Then:

$$E[Y(t)] = \frac{e^{\mu_2+\frac{\sigma_2^2}{2}}\left[1-\text{erf}\left(\frac{\ln t-(\mu_2+\sigma_2^2)}{\sigma_2\sqrt{2}}\right)\right]}{1-\text{erf}\left[\frac{\ln t-\mu_2}{\sigma_2\sqrt{2}}\right]} - t$$

where erf is error function and μ_2 and σ_2 are mean and variance of X_2, respectively. Since in this case, as well in general, the expected value of $Y(t)$ depends on the value of t, then, denoting by $d_{ACB}(t)$ the probability density of the time difference between the meeting of node A with C to time of meeting of node A with B before A meets C, we get:

$$E[Y] = \int_{t=0}^{\infty} \frac{\int_{y=t}^{\infty}(1-D_2(y))dy}{1-D_2(t)} d_{ACB}(t)dt$$

Clearly, $E[Y]$ depends on probability density function $d_{ACB}(t)$ that is defined by the correlation between the meetings of node A with nodes B and C. Due to the nature of DTNs, and also the possible cyclic behavior of nodes [27], there is often a repeating pattern between the meetings of node A with node B and C, creating a strong correlation. Consequently, $E[Y]$ strongly depends on the aforementioned correlation.

Given A, to see the distribution of $\tau_A(C|B)$ according to different B and C nodes, we computed the average conditional intermeeting time maps (C-Map) of the most popular nodes (having the highest total meeting count with other nodes) in three different datasets. In Figure 3, we show these results. For each dataset, the upper plot shows the 3D view of $\tau_A(C|B)$ and the lower plot shows the contour plot displaying the isolines of $\tau_A(C|B)$. The darker the color in the plots, the smaller the $\tau_A(C|B)$ values. Interestingly, the diagonals and the A^{th} row and column in the plots are the darkest places because $\tau_A(C|B)$ values are assumed zero in these cases. Moreover, if there is no instance of the case in which node A meets node C after it meets node B, we assumed $\tau_A(C|B) = -1$.

If there were no correlation between the meetings of node A with other nodes, $\tau_A(C|B)$ would have been the same for all B's other than A or C and entire rows would have been of the same color in contour plots. In contrast, we observe different colors within the rows, demonstrating that the meetings of node A with different nodes are not independent of each other. Indeed this result matches with real world scenarios. For example, consider the meetings of a man who goes from home to work every morning. After meeting his family members (while leaving home), he meets later his office friends. Yet, on the way to his office, he meets the security guard at the gate of his workplace a few moments before meeting his office friends. In other words, the meetings of the man with his office friends is correlated with his meetings with security guard.

IV. Proposed Algorithms

In this section, we present two different utilization of conditional intermeeting time in existing DTN routing algorithms. In the first part, we look into the shortest path based routing algorithms and propose to use conditional shortest paths to route messages. Then, in the second part, we propose to revise message forwarding decision of metric-based forwarding algorithms by including the conditional intermeeting time.

A. Shortest Path based Routing

1) Overview: Shortest path routing (SPR) protocols for DTNs are based on the designs of routing protocols for traditional networks. The links between each pair of nodes are assigned a cost and messages are forwarded over the shortest paths between the source and the destination. Furthermore, the dynamic nature of DTNs is also addressed in these designs. Two of the common metrics used to define the link cost are minimum expected delay (MED [14]) and minimum estimated expected delay (MEED [14]). These metrics compute the expected waiting time plus the transmission delay between each pair of nodes. However, while the former uses the future contact schedule, the latter uses only observed contact history.

In this paper, we are interested in neither the suitability of SPR algorithms for DTNs nor the scalability and complexity of their designs as the papers introducing the original algorithms include elaborate discussions of these issues. In this paper, we focus on the enhancements of the performance of SPR algorithms achieved by utilizing our metric, conditional intermeeting time, rather than using standard intermeeting time. To this end, in the rest of this section, we show the necessary changes to the current designs of SPR algorithms.

2) Network Model: We model a DTN as a graph G = (V, E) where the mobile nodes are represented by vertices (V) and the possible connections between these nodes are represented by the edges ($E = E_u \cup E_b$), which, unlike previous DTN graph models, allows for multiple and both unidirectional (E_u) and bidirectional (E_b) edges between the nodes. The neighbors of a node i are denoted by $N(i)$ and the edge sets are given as follows:

$$\begin{aligned} E_b &= \{(i,j) \mid \forall j \in N(i)\} \text{ where, } w(i,j) = \tau_i(j) = \tau_j(i) \\ E_u &= \{(i,j) \mid \forall j \neq k \in N(i)\} \text{ where, } w(i,j) = \tau_i(j|k) \end{aligned}$$

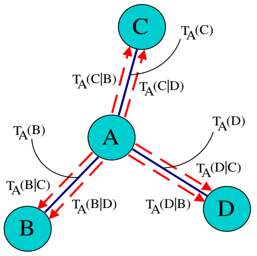


Fig. 4: A sample DTN graph with four nodes and nine edges.

Note that there can be multiple unidirectional edges (E_u) between any two nodes but these edges differ from each other in terms of their weights ($w(i,j)$) and the corresponding third node which is the reference point used while computing the conditional intermeeting time. In Figure 4, we illustrate a sample DTN graph with four nodes and nine edges. There are three bidirectional edges with weights of standard intermeeting times and six unidirectional edges with weights of conditional intermeeting times.

3) Conditional Shortest Path Routing: Our algorithm basically finds conditional shortest paths (CSP) for each source-destination pair and routes the messages over these paths. We define the CSP from a node n_0 to a node n_d as follows:

$$\begin{aligned} CSP(n_0, n_d) &= \{n_0, n_1, \ldots, n_{d-1}, n_d \mid \Re_{n_0}(n_1|t) + \\ &\quad \sum_{i=1}^{d-1} \tau_{n_i}(n_{i+1}|n_{i-1}) \text{ is minimized.}\} \end{aligned}$$

Here, t represents the time that has passed since the last meeting of n_0 with n_1 and $\Re_{n_0}(n_1|t)$ is the expected residual time to the next meeting of n_0 and n_1 given that they have not met in the last t time units. $\Re_{n_0}(n_1|t)$ can be computed as in [19] with parameters of distribution representing the intermeeting time between n_0 and n_1. It can also be computed in a discrete manner from the observed intermeeting times of n_0 and n_1. Assume that n_0 observed k intermeeting times with n_1 in the past. Let $\tau^1_{n_0}(n_1)$, $\tau^2_{n_0}(n_1)$,...$\tau^k_{n_0}(n_1)$ denote these values. Then, discrete computation of $\Re_{n_0}(n_1|t)$ can be defined formally as follows:

$$\Re_{n_0}(n_1|t) = \frac{\sum_{s=1}^{k} f^s_{n_0}(n_1)}{|\{\tau^s_{n_0}(n_1) \geq t\ \}|} \text{ where,}$$

$$f^s_{n_0}(n_1) = \begin{cases} \tau^s_{n_0}(n_1) - t & \text{if } \tau^s_{n_0}(n_1) \geq t \\ 0 & \text{otherwise} \end{cases}$$

If none of the k observed intermeeting times is bigger than t (this case occurs less likely as the the contact history grows), $\Re_{n_0}(n_1|t)$ becomes 0, which is a good approximation.

Next, we give an example to show the benefit of CSP over SP. Consider the DTN shown in Figure 5. The weights of edges (A, C) and (A, B) show the expected residual time of node A with nodes C and B, respectively, in both graphs. The weights of edges (C, D) and (B, D) are different in both graphs. While in the left graph, they show the average intermeeting times of nodes C and B with D respectively, in the right graph, they show the average conditional intermeeting times of the same

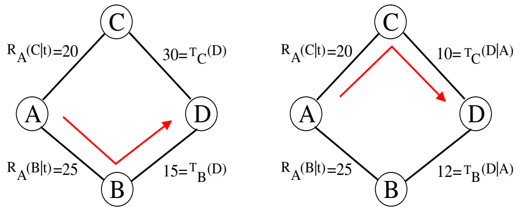


Fig. 5: An example case where CSP can be different than SP.

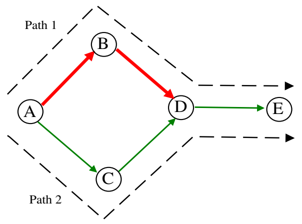


Fig. 6: Path 2 may have smaller conditional delay than path 1 even though CSP from A to D is through B.

nodes with D relative to their meeting with node A. From the left graph, we conclude that SP(A, D) is (A, B, D). Thus, it is expected that on average a message from node A will be delivered to node D in 40 time units. However this may not be the actual shortest delay path. As the weight of edge (C, D) states in the right graph, node C can have a smaller conditional intermeeting time (than the standard intermeeting time) with node D assuming that it has met node A. In other words, node C understands its faster transfer capability of messages (received from node A) to node D. Hence, in the right graph, CSP(A, D) is (A, C, D) with the path cost of 30 time units.

A node-specific view of the network using the aforementioned network model is formed at each node and the standard and conditional intermeeting times of other nodes are collected via epidemic link state protocol as it is described in original study [14]. However, once the weights are known, it is not as easy to find CSPs as it is to find SPs. Consider Figure 6 where the CSP(A, E) follows path 2 and CSP(A, D) follows (A, B, D). This situation is likely to happen in a DTN, if $\tau_D(E|B) \geq \tau_D(E|C)$ is satisfied. Running Dijkstra's or Bellman-Ford algorithm on the current graph structure cannot detect such cases and concludes that CSP(A, E) is (A, B, D, E). Therefore, to obtain the correct CSPs for each source destination pair, we propose the following transformation on the current graph structure.

Given a graph $G = (V, E)$, we obtain a new graph $G' = (V', E')$ where:

$$\begin{aligned} V' &\subseteq V \times V \text{ and } E' \subseteq V' \times V' \text{ where,} \\ V' &= \{(i_j) \mid \forall j \in N(i)\} \text{ and } E' = \{(i_j, k_l) \mid i = l\} \\ &\text{where, } w'(i_j, k_l) = \begin{cases} \tau_i(k|j) & \text{if } j \neq k \\ \tau_i(k) & \text{otherwise} \end{cases} \end{aligned}$$

Remark that the edges in E_b (in G) are made directional in G'. Also the unidirectional edges (E_u) between the same pair of nodes in G are separated in E'. This graph transformation keeps all the historical information that conditional intermeeting times require and also keeps only the paths with a valid history. For example, if a path A, B, C, D is chosen, then an edge like (C_D, D_A) cannot be chosen because of the edge settings in the graph. Hence, only the correct τ values will be added to the path calculation. To solve the conditional shortest path problem however, we add one vertex for source S (apart from its permutations) and one vertex for destination node D. We also add outgoing edges from S to each vertex $(i_S) \in V'$ with weight $\Re_S(i|t)$. Furthermore, for the destination node, D, we add only incoming edges from each vertex $i_j \in V'$ with weight $\tau_i(D|j)$ and from S with weight $\Re_S(D|t)$.

In Figure 7, we show a sample transformation of a clique of four nodes to the new graph structure. In the initial graph, all mobile nodes A to D meet with each other, and we set the source node to A and destination node to D (we did not show the directional edges in original graph for brevity). Note that we set any path to begin with A on transformed graph G', but we also put the permutations of A, B and C with each other.

Running Dijkstra's shortest path algorithm on G' given the source node S and destination node D will give shortest conditional path. In G', $|V'| = O(|V|^2)$ and $|E'| = O(|V^3|) = |E|^{3/2}$, and therefore Dijkstra's algorithm will run in $O(|V|^3)$ (with Fibonacci heaps) while computing the original shortest paths (with standard intermeeting times) takes $O(|V|^2)$.

Using conditional intermeeting times instead of standard intermeeting times only requires (over original design) extra space to store the conditional intermeeting times and additional processing as complexity of running Dijkstra's algorithm increases from $O(|V|^2)$ to $O(|V|^3)$. We believe that in current DTNs, wireless devices have enough storage and processing power not to be unduly taxed with such an increase. Moreover, to lessen the burden of collecting and storing link weights, an asynchronous and distributed version of the Bellman-Ford algorithm can be used, as described in [28].

B. *Metric-based Forwarding Algorithms*

1) Overview: In DTNs, a common routing method is to forward the message to the encountered node that is more likely to meet with destination than the current message carrier. However, making effective forwarding decisions in single-copy based routing in DTNs is a challenging task. When two nodes meet, one of them forwards a message to the other one if it decides that the message will have a higher chance to be delivered to destination at the other node.

In previous work, depending on the observed contact history between nodes, several metrics have been used to define the delivery quality of nodes. Some of the popular ones are encounter frequency [15], time elapsed since last encounter [21] [22], residual time [19] and social similarity [23]-[25]. For example, in Prophet [15], messages are forwarded to the nodes meeting the destination more frequently.

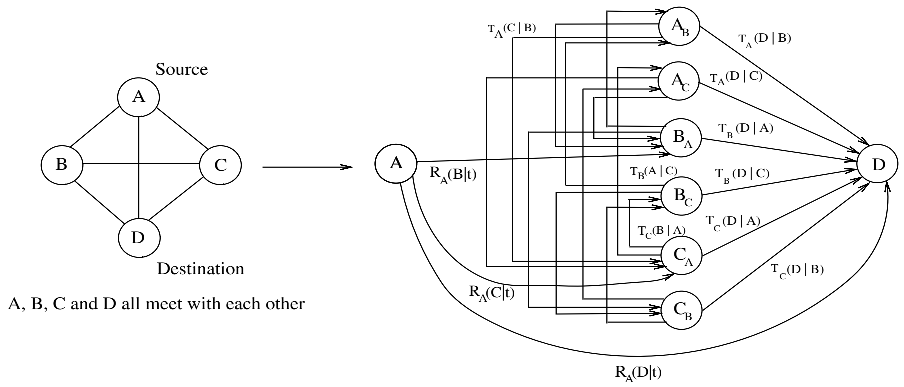


Fig. 7: Graph transformation to solve CSP with 4 nodes where A is the source and D is the destination.

2) Proposed Revision: According to most of the delivery metrics proposed previously, two encountering nodes make the message forwarding decision depending on their individual relations with the destination node. In some algorithms such as [15] [22], transitivity rule is utilized to reflect the effect of other nodes on the delivery quality of a node but such an approach can be applied to all delivery metrics and it does not reflect the metric's own feature.

To make forwarding decisions of these algorithms more effective, thus to improve their performance, we propose to use conditional intermeeting time as an additional delivery metric. That is, when two nodes meet, they will also compare their conditional intermeeting times with destination and if the current carrier of the message learns that other node has also shorter remaining time (according to conditional intermeeting time) to meet the destination than itself, the message is forwarded. At first glance, this additional condition seems to cut down the number of times the message is forwarded so that the probability of delivery will be reduced. However, as simulation results show, the necessary number of hops are preserved and the less beneficial ones are not performed. Therefore, more effective forwarding decisions are made so that the cost of message delivery declines while the delivery ratio and average delay are maintained (in some cases, even the delivery ratio increases and average delay decreases).

V. Simulations

To evaluate the performance of proposed modifications of algorithms, we have built a Java based DTN simulator which uses the traces of real objects. We used traces from three datasets which include the contact times and durations of real objects logged in different DTN environments and we set the network parameters (number of nodes etc.) accordingly.

A. Algorithms in Comparison

In simulations, we compared existing DTN algorithms with their modifications utilizing conditional intermeeting time in their designs. First, we compared Shortest Path Routing (SPR) with Conditional Shortest Path Routing (CSPR) which is described in Section IV-A3. Then, we compared the existing and revised versions of two metric-based DTN routing algorithms: Prophet [15] and Fresh [21]. In Prophet, when two nodes, A and B, meet, A forwards its message to B if and only if its predicted time of delivery to destination D is smaller than B's predicted time of delivery. In Fresh, A forwards the message to B only if B has a more recent meeting with destination D than itself. In the revised versions of these algorithms (we refer to them as C-Prophet and C-Fresh to underline that they use conditional intermeeting time), A forwards the message to B if $\tau_A(D|B) > \tau_B(D|A)$ is also satisfied (in addition to algorithm's own forwarding condition). Although we obtained results (showing performance improvement) with many metric-based algorithms (including [19]), we show only the results of two benchmarking algorithms due to lack of space. However, we give the results obtained by Epidemic Routing [6] since it achieves the optimum delivery ratio and delay (at high cost, however).

B. Data Sets

To evaluate the proposed algorithms, we used traces from the following three data sets.

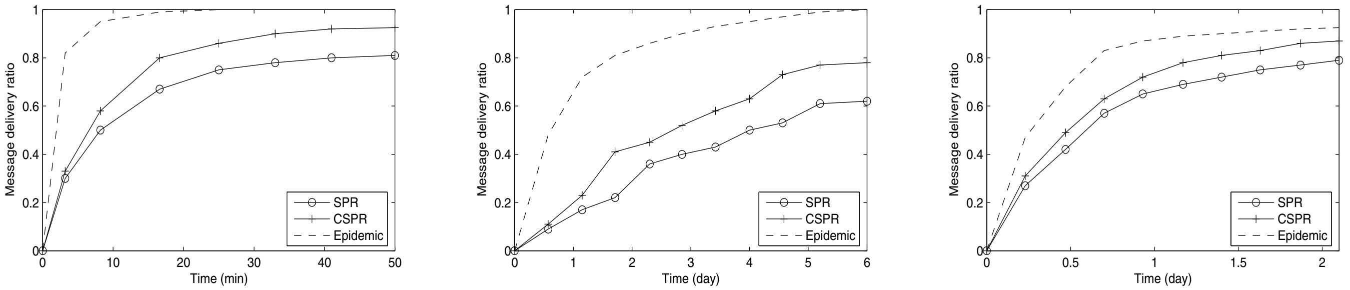
1) RollerNet Dataset [20]: It includes the opportunistic sightings of Bluetooth devices distributed to 62 rollerbladers in the 3 hour roller tour of Paris in August 20, 2006.

2) Cambridge Dataset [30]: This data includes a number of traces of Bluetooth sightings by 36 students from Cambridge University who were asked to carry the iMotes with them at all times for the duration of the experiment that started on October 28, 2005 and ended on December 21, 2005.

3) Haggle Project Dataset [32]: Among the different experiments performed within Haggle Project, we selected the Bluetooth sightings recorded between the iMotes carried by 41 attendants of Infocom'05 Conference held in Miami. Devices were distributed on March 7th, 2005 between lunch time and 5pm and collected on March 10th, 2005 in the afternoon.

C. Performance Metrics

We use the following three metrics to compare the algorithms: message delivery ratio, average cost, and routing efficiency. Delivery ratio is the proportion of messages delivered



(a) RollerNet Traces

(b) Cambridge Traces

(c) Haggle Project Traces

Fig. 8: Comparison of SPR and CSPR: Message delivery ratio vs. time.

to their destinations among all messages generated. Average cost (which is a good indicator of the total power consumption in the network) is the average number of forwardings done per message before delivery. Finally, routing efficiency [33] is defined as the ratio of delivery ratio to the average cost. In the results, we did not give separate plots for delivery delay because they can be obtained from the delivery ratio plots.

D. Results

To collect several routing statistics, we have generated traffic from the traces of three data sets. For each simulation run, after a warm up period, we generated 5000 messages from a random source node to a random destination node at each t seconds. In RollerNet, since the duration of experiment is short, we set $t = 1$s, but for Cambridge and Haggle data sets, we set $t = 1$min and $t = 30$s, respectively. We assume that the nodes have enough buffer space to store every message they receive, the bandwidth is high and the contact durations of nodes are long enough to allow the exchange of all messages between nodes[3]. These assumptions are reasonable in view of today's technology capabilities and are also used commonly in previous studies [29]. Besides, we compare all algorithms in the same conditions. Any change in the current assumptions is expected to affect the performance of compared algorithms in the same way since they use one copy of the message. Moreover, we used a simple MAC model similar to CSMA model. We ran each simulation 10 times with different seeds and in each run we collect statistics by running each algorithm on the same set of messages. All results plotted in figures show the averages of results obtained in all runs (we did not plot the error bars since they were very small).

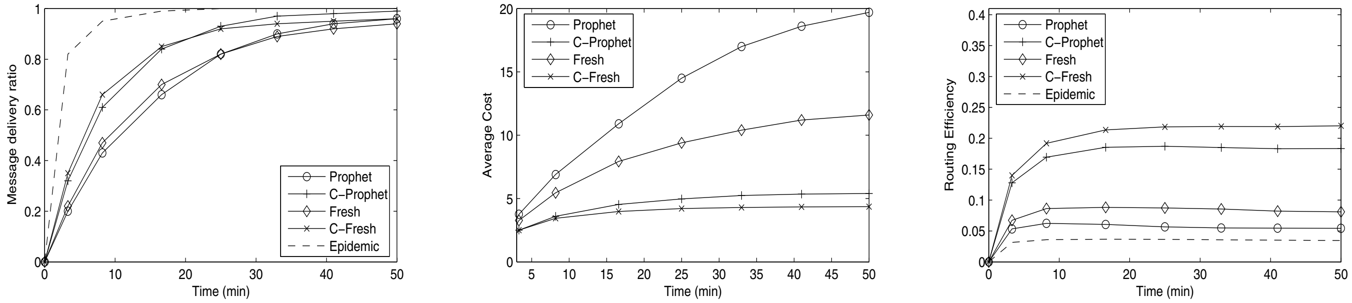
1) Comparison of CSPR and SPR: Figure 8a shows the delivery ratios achieved in CSPR and SPR algorithms with respect to time (i.e. TTL of messages) in RollerNet traces. Clearly, CSPR algorithm delivers more messages to their destinations than SPR algorithm. Moreover, it achieves lower average delivery delay than SPR algorithm. For example, CSPR delivers 80% of all messages after 17 minutes with an average delay of almost 6 minutes, while SPR achieves the same delivery ratio only after 41 minutes and with an average delay of 12 minutes. Moreover, although we did not show it here for brevity, average costs in SPR and CSPR are very close (1.48 and 1.52 respectively) to each other (and much smaller than the average cost in epidemic routing which is around 25).

We also see the better delivery ratios achieved by CSPR algorithm in Cambridge and Haggle traces in Figure 8b and Figure 8c, respectively. In Cambridge traces, after 6 days, CSPR delivers 78% of all messages with an average delay of 2.6 days, however SPR can only deliver 62% of all messages to their destination with an average delay of 3.2 days. Moreover average costs in SPR and CSPR are 1.73 and 1.78 respectively while it is around 16 in epidemic routing. Similarly, in Haggle traces, with an average cost close to each other, CSPR delivers 87% of all messages by the end of simulation whereas SPR can only achieve 78% delivery ratio. To put this in perspective, compared to Epidemic routing with the highest achievable delivery ratio (94% in Haggle traces with current setting), CSPR lost only 7% of messages, while SPR lost 16% so CSPR achieved more than 55% improvement over SPR.

These results show that in the context of routing, conditional intermeeting time provides a better representation of link cost than standard intermeeting time. Therefore, in CSPR, more effective paths with similar average hop counts are selected during the routing of a message towards the destination. Thus, higher delivery ratios with lower end-to-end delays are achieved. In SPR and CSPR algorithms here, we used source-routing [12] and let the messages follow the paths which are decided at the source nodes. We also observed similar results in our simulations with other routing approaches (per-hop and per-contact routing [14]).

2) Comparison of revised and original versions of metric-based algorithms: In Figure 9a, we show the delivery ratios achieved in RollerNet traces. Clearly, the modified algorithms provide higher delivery ratio than the original ones. Moreover, as Figure 9b shows, average cost is lower for the modified versus the original algorithms. For example, C-Prophet delivers 90% of all messages after 23 minutes with average delay of 7.8 minutes and average cost of 4.83 hops. However, the original Prophet reaches the same delivery ratio only after 33 minutes with average delay of 13.5 minutes and 17.02 average cost. A

[3] We also performed simulations with limited resources (e.g. buffer, bandwidth) and obtained results showing better (but different) performance improvement in revised algorithms over original algorithms. However, we present them in the extended version of the paper due to the lack of space here.

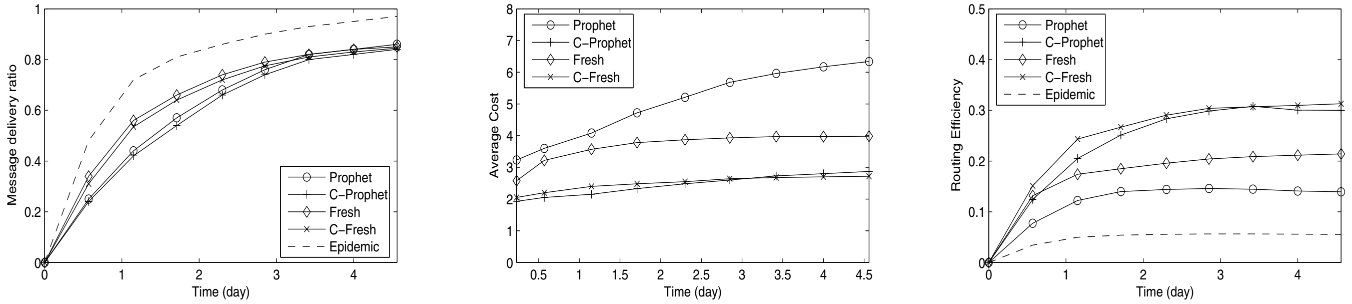


(a) Message delivery ratio vs. time

(b) Average cost vs. time

(c) Routing Efficiency vs. time

Fig. 9: Comparison of metric-based forwarding algorithms using RollerNet traces

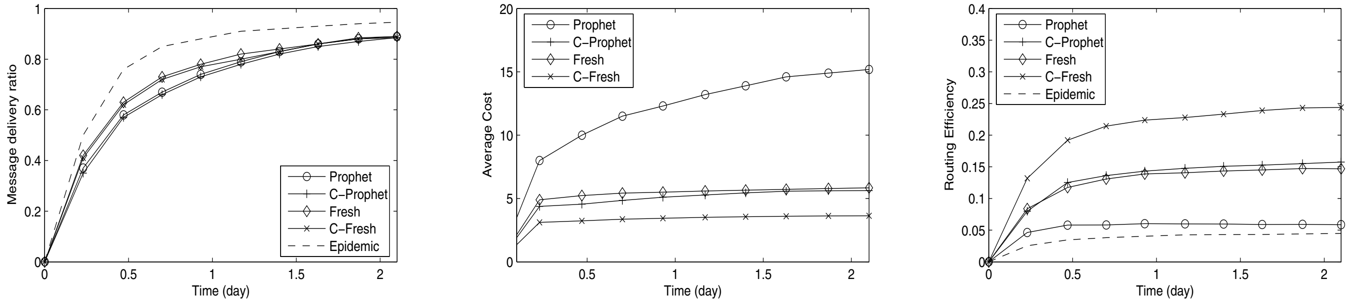


(a) Message delivery ratio vs. time

(b) Average cost vs. time

(c) Routing Efficiency vs. time

Fig. 10: Comparison of metric-based forwarding algorithms using Cambridge traces



(a) Message delivery ratio vs. time

(b) Average cost vs. time

(c) Routing Efficiency vs. time

Fig. 11: Comparison of metric-based forwarding algorithms using Haggle Project traces

similar situation is also observed between C-Fresh and Fresh. Consequently, routing efficiency is increased more than 100%.

When we look at the results obtained from Cambridge and Haggle traces in Figure 10 and Figure 11, we observe a different improvement. As it is seen in Figure 10a and Figure 11a, revised and original versions of algorithms have similar delivery ratios (and therefore similar average delays). However, as Figure 10b and Figure 11b show, average costs in modified versions are lower than they are in original ones. Moreover, in Cambridge traces, the mean hop counts of Prophet, C-Prophet, Fresh and C-Fresh are 5.21, 2.48, 3.83 and 2.53, respectively and in Haggle traces, they are 12.7, 4.98, 5.23 and 3.44. This shows that when conditional intermeeting time is used as an additional delivery metric, the nodes choose their next hops more effectively so that the cost is decreased while still keeping the original delivery ratio. Therefore, again more than 100% gain is achieved in routing efficiency. From the above results, we see the benefit of conditional intermeeting times in metric-based forwarding algorithms clearly. Moreover, we also observe that in different environments, the improvements obtained thanks to utilization of conditional intermeeting times can be different. In RollerNet traces, we observed improvement in all metrics. However, in Cambridge traces we observed an improvement only in average cost, thus, in routing efficiency. From our initial analysis of the traces, we conclude that this is caused by the repetitive contacts between objects in RollerNet traces. That is, as it is stated in [20], during the tour, fluctuations in the motion of the rollerbladers cause a typical accordion phenomenon (the topology expands and shrinks with time). When the topology shrinks, nodes are in contact with each other, but when the topology expands, they are disconnected. This property in the contacts of nodes creates a cyclic behavior and lets the proposed algorithms perform better. However, in Cambridge and Haggle traces, the repetitive behavior of node meetings is not clearly detected. From these results, we conclude that the benefit of using conditional intermeeting times in metric based algorithms becomes more pronounced in the environments in which the repetitive motion of nodes is clearly observed. However, even in the environments where this is not the case, average cost of routing can be decreased and routing efficiency can be improved remarkably thanks to use of conditional intermeeting times.

VI. Conclusion and Future Work

In this paper, we focused on the routing problem in delay tolerant networks (DTN). First, inspired by the results of the recent studies showing that intermeeting times between nodes are not memoryless and the motion patterns of mobile nodes are frequently repetitive, we introduced a new metric called conditional intermeeting time which is the average time that passes from the time a node meets with a neighbor node until the time it meets another one. Next, we presented an analysis of this metric showing why it can be beneficial in proper representation of node relations. Then, we looked at the effects of this metric on existing DTN routing algorithms. To this end, we modified their current designs using conditional intermeeting time. Finally, through real-trace-driven simulations, we evaluated the modified algorithms and demonstrated the superiority of them over original ones.

As a future work of our study, we would like to extend the definition of conditional intermeeting time by using more meetings from the contact history. For instance, assume that a node A wants to compute its conditional intermeeting time with a node B after the time it has met another node C. Here, we want to differentiate the following two cases which are considered together in our current approach; when node A has met node D before node C and when node A has met node E before node C. To apply this algorithm in our work, we plan to use probabilistic context free grammars (PCFG) and utilize the construction algorithm presented in [34].

References

[1] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, *Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet*, in Proceedings of ACM ASPLOS, 2002.

[2] *Disruption tolerant networking*, http://www.darpa.mil/ato/solicit/DTN/.

[3] J. Ott and D. Kutscher, *A disconnection-tolerant transport for drive-thru internet environments*, in Proceedings of IEEE INFOCOM, 2005.

[4] *Delay tolerant networking research group*, http://www.dtnrg.org.

[5] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, *MaxProp: Routing for Vehicle-Based Disruption- Tolerant Networks*, in Proc. IEEE Infocom, April 2006.

[6] A. Vahdat and D. Becker, *Epidemic routing for partially connected ad hoc networks*, Duke University, Tech. Rep. CS-200006, 2000.

[7] T. Spyropoulos, K. Psounis,C. S. Raghavendra, *Efficient routing in intermittently connected mobile networks: The multi-copy case*, IEEE/ACM Transactions on Networking, 2008.

[8] E. Bulut, Z. Wang, and B. Szymanski, *Cost-Effective Multi-Period Spraying for Routing in Delay Tolerant Networks*, to appear in IEEE/ACM Transactions on Networking, 2010.

[9] Y. Wang, S. Jain, M. Martonosi, and K. Fall, *Erasure coding based routing for opportunistic networks*, in Proceedings of ACM SIGCOMM workshop on Delay Tolerant Networking (WDTN), 2005.

[10] E. Bulut, Z. Wang, B. Szymanski, *Cost Efficient Erasure Coding based Routing in Delay Tolerant Networks*, in Proceedings of ICC 2010.

[11] I. Psaras, L. Wood and R. Tafazolli, *Delay-/Disruption-Tolerant Networking: State of the Art and Future Challenges*, Technical Report, University of Surrey, UK, 2010.

[12] S. Jain, K. Fall, and R. Patra, *Routing in a delay tolerant network*, in Proceedings of ACM SIGCOMM, Aug. 2004.

[13] T. Spyropoulos, K. Psounis,C. S. Raghavendra, *Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks*, ACM SIGCOMM Workshop, 2005.

[14] E. P. C. Jones, L. Li, and P. A. S. Ward, *Practical routing in delay tolerant networks*, in Proceedings of ACM SIGCOMM workshop on Delay Tolerant Networking (WDTN), 2005.

[15] A. Lindgren, A. Doria, and O. Schelen, *Probabilistic routing in intermittently connected networks*, SIGMOBILE Mobile Computing and Communication Review, vol. 7, no. 3, 2003.

[16] 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 Proceedings of INFOCOM, 2006.

[17] T. Karagiannis, J. Boudec, and M. Vojnovic, *Power Law and Exponential Decay of Inter Contact Times Between Mobile Devices*, in Proceedings of MobiCom, 2007.

[18] X. Zhang, J. F. Kurose, B. Levine, D. Towsley, and H. Zhang, *Study of a Bus-Based Disruption Tolerant Network: Mobility Modeling and Impact on Routing*, in Proceedings of ACM MobiCom, 2007.

[19] S. Srinivasa and S. Krishnamurthy, *CREST: An Opportunistic Forwarding Protocol Based on Conditional Residual Time*, in Proceedings of SECON, 2009.

[20] P. U. Tournoux, J. Leguay, F. Benbadis, V. Conan, M. Amorim, J. Whitbeck, *The Accordion Phenomenon: Analysis, Characterization, and Impact on DTN Routing*, in Proceedings of Infocom, 2009.

[21] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli, *Age Matters: Efficient Route Discovery in Mobile Ad Hoc Networks Using Encounter Ages*, in Proceedings of ACM MobiHoc, 2003.

[22] T. Spyropoulos, K. Psounis, and C. Raghavendra, *Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility*, in Proceedings of IEEE PerCom, 2007.

[23] E. Daly and M. Haahr, *Social network analysis for routing in disconnected delay-tolerant manets*, in Proceedings of ACM MobiHoc, 2007.

[24] P. Hui, J. Crowcroft and E. Yoneki, *BUBBLE Rap: Social Based Forwarding in Delay Tolerant Networks*, in Proceedings of ACM MobiHoc, 2008.

[25] E. Bulut, Z. Wang and B. Szymanski, *Impact of Social Networks in Delay Tolerant Routing*, in Proceedings of Globecom, 2009.

[26] T. Spyropoulos, K. Psounis,C. S. Raghavendra, *Performance Analysis of Mobility-assisted Routing*, MobiHoc, 2006.

[27] C. liu, J. Wu and I. Cardei *Message Forwarding in Cyclic Mobispace: the Multi-copy case*, in Proceedings of the Sixth IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS), 2009.

[28] D. Bertsekas, and R. Gallager, *Data networks (2nd ed.)*, 1992.

[29] C. Liu and J. Wu, *An Optimal Probabilistically Forwarding Protocol in Delay Tolerant Networks*, in Proceedings of MobiHoc, 2009.

[30] J. Leguay, A. Lindgren, J. Scott, T. Friedman, J. Crowcroft and P. Hui, *CRAWDAD data set upmc/content (v. 2006-11-17)*, downloaded from http://crawdad.cs.dartmouth.edu/upmc/content, 2006.

[31] Y. Wang, P. Zhang, T. Liu, C. Sadler and M. Martonosi, *http://crawdad.cs.dartmouth.edu/princeton/zebranet*, CRAWDAD data set princeton/zebranet (v. 2007-02-14), 2007.

[32] A European Union funded project in Situated and Autonomic Communications, *www.haggleproject.org*.

[33] J. M. Pujol, A. L. Toledo, and P. Rodriguez, *Fair routing in delay tolerant networks*, in Proc. IEEE INFOCOM, 2009.

[34] S. Geyik, and B. Szymanski, *Event Recognition in Sensor Networks by Means of Grammatical Inference*, in Proc. of INFOCOM, Brazil, 2009.