International Journal of Electronics Communication and Computer Engineering
Volume 4, Issue (6) NCRTCST-2013, ISSN 2249–071X

National Conference on Recent Trends in Computer Science and Technology (NCRTCST)-2013

Determining Conditional Intermeeting Time in DTMN Networks

Sindhu Vegesna
Asst. Professor,
Dept. of CSE, CMRCET

I. Haripriya
Asst. Professor,
Dept.of CSE, CMRCET

G. Ravi Kumar
Associate Professor,
Dept. of CSE, CMRCET

K. Venkateswara Rao
Asst. Professor,
Dept. of CSE, CMRCET

***Abstract* – In this paper, we discussed about delay tolerant mobile networks in conditional intermeeting time. It is a new metric in DTMN. Normally 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 history. And also the paper proposed metric on the shortest path based routing designed for delay tolerant networks. The paper proposes Conditional Shortest Path Routing protocol that routes the messages over conditional shortest paths in which the cost of links between nodes is defined by conditional intermeeting times rather than the conventional intermeeting times. Through trace-driven simulations, we demonstrate the algorithm called CSPR. It achieves higher delivery rate and lower end-to-end delay compared to the shortest path based routing protocols that use the conventional intermeeting time as the link metric.**

***Keywords* – Delay Tolerant Mobile Networks (DTMN), Opportunistic Forwarding, Shortest Path Routing, CSPR.**

I. Introduction

Delay-tolerant networks (DTNs) have the potential to connect devices and areas of the world that are under-served by current networks. Delay tolerant networks are characterized by the spasmodic connectivity between their nodes and therefore the lack of stable end-to-end paths from source to destination. Since the future node connections are mostly unknown in these networks **opportunistic forwarding is** used to deliver messages. However, making effective forwarding decisions using only the network characteristics extracted from contact history is a challenging problem. A critical challenge for DTNs is determining routes through the network without ever having an end-to-end connection, or even knowing which “routers” will be connected at any given time. Routing algorithms in DTN’s utilize a paradigm called store-carry-and-forward. When a node receives a message from one of its contacts, it stores the message in its buffer and carries the message until it encounters another node which is at least as useful (in terms of the delivery) as itself. Then the message is forwarded to it.

II. Related Work

Based on this paradigm, several Routing algorithms with different objectives (high delivery rate etc.) and different routing techniques (single-copy, multi-copy, erasure coding based etc.) have been proposed recently. However, some of these algorithms used unrealistic assumptions, such as the existence of predict which provide future contact times of nodes. Yet, there are also many algorithms based on realistic assumption of using only the contact history of nodes to route messages opportunistically. Recent studies on routing problem in DTN’s have focused on the analysis of real mobility traces (human, vehicular etc.). Different traces from various DTN environments are analyzed and the extracted characteristics of the mobile objects are utilized on the design of routing algorithms for DTN’s. From the analysis of these traces performed in previous work, we have made two key observations. First, rather than being memory less, the pair wise intermeeting times between the nodes usually follow a log-normal distribution Therefore, future contacts of nodes become dependent on the previous contacts. Second, the mobility of many real objects are non-deterministic but cyclic. Hence, in a cyclic MobiSpace, if two nodes were often in contact at a particular time in previous cycles, then they will most likely be in contact at around the same time in the next cycle.

To show the benefits of the proposed metric, we adopted it for the shortest path based routing algorithms designed for DTN’s. We propose *conditional shortest path routing* (CSPR) protocol in which average conditional intermeeting times are used as link costs rather than standard[2] intermeeting times and the messages are routed over conditional shortest paths (CSP). We compare CSPR protocol with the existing shortest path (SP) based routing protocol through real-trace-driven simulations. The results demonstrate that CSPR achieves higher delivery rate and lower end-to-end delay compared to the shortest path based routing protocols. This show well the conditional intermeeting time represents inter-node link costs (in the context of routing) and helps making effective forwarding decisions while routing a message.

III. Conditional Shortest Path Routing

A. Overview

Shortest path routing protocols for DTN’s are based on the designs of routing protocols for traditional networks. Messages are forwarded through the shortest paths between source and destination pairs according to the costs assigned to links between nodes. Furthermore, the dynamic nature of DTN’s is also considered in these designs. Two common metrics used to define the link costs are minimum expected delay (MED) and minimum

International Journal of Electronics Communication and Computer Engineering
Volume 4, Issue (6) NCRTCST-2013, ISSN 2249–071X

estimated expected delay (MEED). They 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.

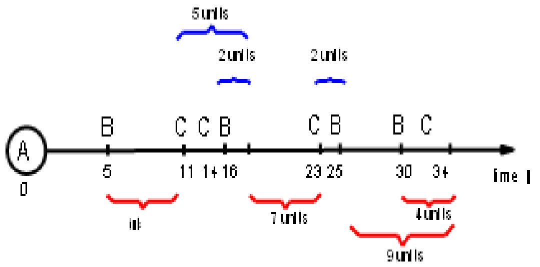

Fig.2. Sample meeting times of node A with nodes B and C. While the values in the upper part are used in the computation of T_A (B|C), the values in the lower part are used in the computation of T_A (C |B).

Routing decisions can be made at three different points in an SP based routing: i) at source, ii) at each hop, and iii) at each contact. In the first one (source routing), SP of the message is decided at the source node and the message follows that path. In the second one (per-hop routing), when a message arrives at an intermediate node, the node determines the next hop for the message towards the destination and the message waits for that node. Finally, in the third one (per-contact routing), the routing table is recomputed at each contact with other nodes and the forwarding decision is made accordingly. In these algorithms, utilization of recent information increases from the first to the last one so that better forwarding decisions are made; however, more processing resources are used as the routing decision is computed more frequently.

B. Network Model

We model a DTN as a graph G = (V, E) where the vertices (V) are mobile nodes and the edges (E) represent the connections between these nodes. However, different from previous DTN network models, we assume that there may be multiple unidirectional (E_u) and bidirectional (E_b) edges between the nodes. The neighbors of a node i are denoted with N(i) and the edge sets are given as follows:

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

The above definition of Eu allows for multiple unidirectional edges between any two nodes. However, these edges differ from each other in terms of their weights and the corresponding third node. This third node indicates the previous meeting and is used as a reference point while defining the conditional intermeeting time (weight of the edge). In Figure 3, we illustrate a sample DTN graph with four nodes and nine edges. Of these nine edges, three are bidirectional with weights of standard intermeeting times between nodes, and six are unidirectional edges with weights of conditional intermeeting times.

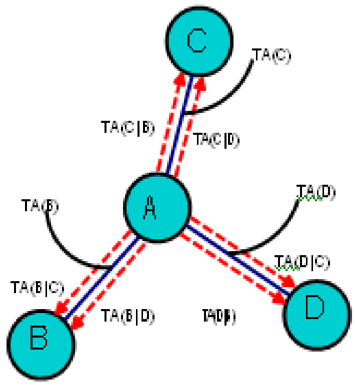

Fig.3. The graph of a sample DTN with four nodes and nine edges in total.

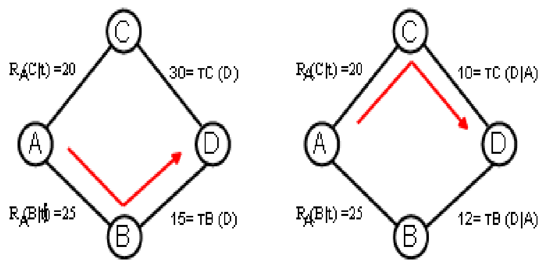

Fig.4. The shortest path from a source to destination node can be different when conditional intermeeting times are used as the weights of links in the network graph.

C. 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:

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

Here, t represents the time that has passed since the last meeting of node n_0 with n_1 and $\Re n_0$ (n1|t) is the expected residual time for node n0 to meet with node n_1 given that they have not met in the last t time units. $\Re n_0$ (n1|t) can be computed as in 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 contact history of n_0 and n_1. Assume that node i observed d intermeeting times with node j in its past. Let $T_i^1(j)$, $T_i^2(j)$,. . .$T_i^d(j)$ denote these values

International Journal of Electronics Communication and Computer Engineering
Volume 4, Issue (6) NCRTCST-2013, ISSN 2249–071X

$$\Re_i(j|t) = \frac{\sum_{k=1}^{d} f_i^k(j)}{\|\tau_i^k(j) \ge t\|} \text{ where,}$$

$$f_i^k(j) = \begin{cases} \tau_i^k(j) - t & \text{if } \tau_i^k(j) \ge t \\ 0 & \text{otherwise} \end{cases}$$

Here, if none of the d observed intermeeting times is bigger than t (this case occurs less likely as the contact history grows), a good approximation can be to assume $\Re_i(j|t) = 0$.

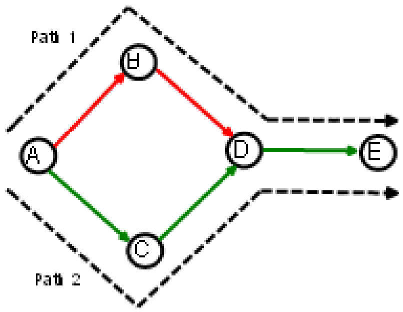

Fig.5. Path 2 may have smaller conditional delay than path 1even though the CSP from node A to node D is through node B.

We will next provide an example to show the benefit of CSPover SP. Consider the DTN illustrated in Figure 4. 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. But 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 nodes with D relative to their meeting with node A. From the left graph, we conclude that SP (A, D) follows (A, B, D). Hence, 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. This provides node C with a faster transfer of the message to node D after meeting node A. Hence, in the right graph, CSP (A, D) is (A, C, D) with the path cost of 30 time units.

Each node forms the aforementioned network model and collects the standard and conditional intermeeting times of other nodes between each other through epidemic link state protocol. However, once the weights are known, it is not as easy to find CSP's as it is to find SP's. Consider Figure 5 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 $T_D(E|B) \ge T_D(E|C)$ 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, and E). Therefore, to obtain the correct CSP's for each source destination pair, we propose the following transformation on the current graph structure.

Given a DTN graph G = (V, E), we obtain a new graph

$$V' \subseteq V \times V \text{ and } E' \subseteq V' \times V' \text{ where,}$$

$$V' = \{(i_j) \mid \forall j \in N(i)\} \text{ and } E' = \{(i_j, k_l) \mid i = l\}$$

$$\text{where, } w(i_j, k_l) = \begin{cases} T_i(k|j) & \text{if } j = k \\ T_i(k) & \text{otherwise} \end{cases}$$

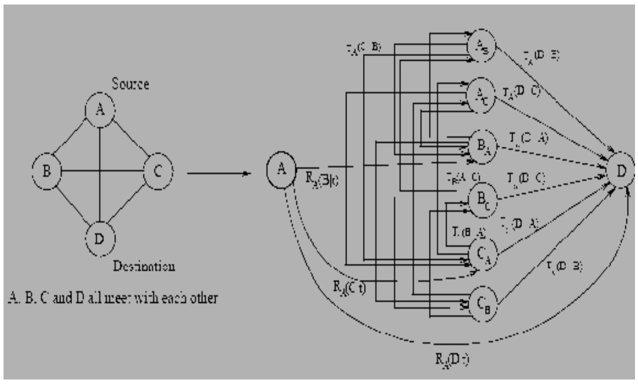

Fig.6. Graph Transformation to solve CSP with 4 Nodes where A is the source and D the destination node.

Note that the edges in E_b (in G) are made directional in G' and the edges in E_u between the same pair of nodes 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, for a path A, B, C, D in G, an edge like (CD, DA) in G′ cannot be chosen because of the edge settings in the graph. Hence, only the correct T values will be added to the path calculation. To solve the CSP 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 (is) ∈ V′ with weight $\Re_S(i|t)$. Furthermore, for the destination node, D, we add only incoming edges from each vertex $\mathbf{i}_j \in V'$ with weight $T_i(D|j)$.

In Figure 6, 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 the original graph for brevity). It can be seen 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 D will give CSP. In G′, $|V'| = O(|V|^2)$ and $|E'| = O(|V^3|) = |E|^{3/2}$. Therefore Dijkstra's algorithm will run in $O(|V|^3)$ (with Fibonacci heaps) while computing regular shortest paths (where edge costs are standard intermeeting times) takes $O(|V|^2)$.

The focus of this paper is an improvement of the current design of the Shortest Path (SP) based DTN routing algorithms. Therefore we leave the elaborate discussion of some other issues in SP based routing (complexity, scalability and routing type selection) to the original studies. Using conditional instead of standard intermeeting times requires 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 DTN's, 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.

IV. Simulations

To evaluate the performance of our algorithm, we have built a discrete event simulator in Java. In this section, we describe the details of our simulations through which we compare the proposed ***Conditional Shortest Path Routing*** (CSPR) algorithm with standard ***Shortest Path Routing*** (SPR). Moreover, in our results we also show the performance of upper and lower performance boundaries with Epidemic Routing and Direct Delivery.

V. Conclusion and Future Work

In this paper, we introduced a new metric called conditional intermeeting time inspired by the results of the recent studies showing that nodes' intermeeting times are not memory less and that motion patterns of mobile nodes are frequently repetitive. Then, we looked at the effects of this metric on shortest path based routing in DTN's. For this purpose, we updated the shortest path based routing algorithms using conditional intermeeting times and proposed to route the messages over conditional shortest paths. Finally, we ran simulations to evaluate the proposed algorithm and demonstrated the superiority of CSPR protocol over the existing shortest path routing algorithms. In future work, we will look at the performance of the proposed algorithm in different data sets to see the effect of conditional intermeeting time in different environments. Moreover, we will consider extending our CSPR algorithm by using more information (more than one known meetings) from the contact history while deciding conditional intermeeting times. For this, we plan to use probabilistic context free gram-mars (PCFG) and utilize the construction algorithm. Such a model will be able to hold history information concisely, and provide further generalizations for unseen data.

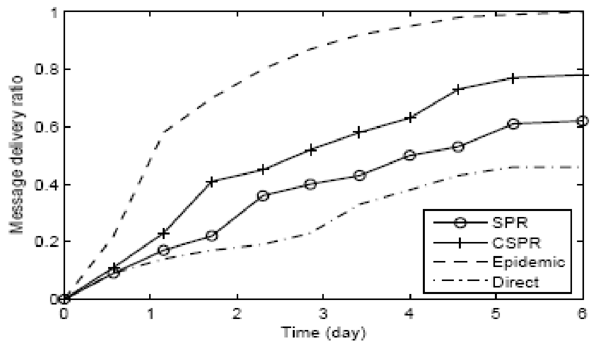

Fig.8. Message delivery ratio vs. time in Cambridge traces.

VI. References

[1] Eyuphan Bulut, Sahin Cem Geyik, Boleslaw K. Szymanski, *Conditional Shortest Path Routing in Delay Tolerant Networks*, in Proceedings of IEEE international Symposium on a world of Wireless Mobile and Multimedia Networks, WoWMoM 2010, Montreal, Canada, June 14-17,2010, pp. 1-6.

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

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

[4] 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.