RESEARCH ARTICLE OPEN ACCESS

Delay Tolerant Network Routing Nature – A Study

V. Prema Tulasi[1], K. Satyanarayana[2], Revuri VARUN[3], VARUN Maramraju[4]
[1]Assistant Professor, Dept Of Cse, Nalla Narasimha Reddy Educational Society's Group Of Institutions, Hyderabad, Ap, India.
[2]Assistant Professor, Dept Of Cse,Sphoorthy Engg College, Hyderabad, Ap, India
[3,4]Assistatnt Professor,Dept Of Cse, Vaagdevi College Of Engineering,Bollikunta, Warangal, Ap, India

Abstract
Delay-Tolerant network (DTN) is a network in which no simultaneous end-to-end path exists. And the messages delivered in the DTN usually have large delivery latency due to network partition. These special characteristics make DTN routing a challenging problem. 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. This proposes Conditional Shortest Path Routing (CSPR) protocol that route 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.

Keywords: DTN, CSPR, Routing, Network;

I. INTRODUCTION

Routing in delay tolerant networks (DTN) is a challenging problem because at any given time instance, the probability that there is an end-to-end path from a source to a destination is low. Since the routing algorithms for conventional networks assume that the links between nodes are stable most of the time and do not fail frequently, they do not generally work in DTN's. Therefore, the routing problem is still an active research area in DTN's [1-10].

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. Based on this paradigm, several routing algorithms with different objectives (high delivery rate etc.) and different routing techniques (single-copy [1-10], multi-copy [1-10], erasure coding based [1-10] etc.) have been proposed recently. However, some of these algorithms [7] used unrealistic assumptions, such as the existence of oracles which provide future contact times of nodes. Yet, there are also many algorithms (such as [1-10]) based on realistic assumption of using only the contact history of nodes to route messages opportunistically.

II. LITERATURE

Work on DTN networks shows that it is possible to automatically route in networks, even when nodes are mobile and the link quality varies. There is a huge body of work on routing protocols [11-22] and metrics [11-22] for this environment. However, these protocols and metrics find end-to-end paths, and do not support communication between nodes in different network partitions. Recent studies on routing problem in DTN's have focused on the analysis of real mobility traces (human [11-22], vehicular [11-22] 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. An approach that uses a single copy of each message is presented by Jain *et al.* [11-22]. They assume that the contact schedule is completely known in advance, and use this knowledge to create a number of routing metrics. Their results show that the efficiency and performance increases with the amount of information used for the metric. The weakness of this approach is that each node must have access to accurate schedule data. To provide this information, the routing must be manually configured with the contact schedules, which must be repeated each time the schedule changes. Handorean *et al.* explore alternatives for distributing connectivity information, but they still assume that each node knows its own connectivity perfectly [11-22]. 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 [11-22]. 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 [11-22], 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.

III. ROUTING

In wireless sensor networks or in networking sending the data or information in packets from source sender to receiver destination, Sending data from sender side system to receiver side routing is classified into static and dynamic routing.

Static Routing: In this routing, information of all networks must be entered manually by the administrator on all routers. As routing data is entered by the administrator there is no requirement to perform any metric calculations to determine paths. Advantages of static routing are: Doesn't exchange any routing tables hence it doesn't consume any bandwidth. Again the bandwidths which may be consumed in dynamic updates are saved. Low processing power may suffice; hence we can use cheaper routers. Higher Security as there is no information being regarding routing on the connecting links. Draw backs of static routing are each remote network information is written manually. Reconfiguration is problematic and tedious if there is a change in the topology.

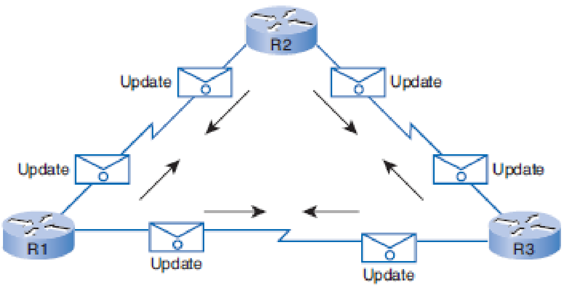

Fig. 1. Dynamic routers

Dynamic Routing: These routing protocols are used to facilitate the exchange of routing information between routers. Routing protocols allow routers to dynamically learn information of all remote networks and automatically add this information to their own routing. Routing protocols determine the best path to each network, which is then added to the routing table. One of the primary benefits of using a dynamic routing protocol is that routers exchange routing information whenever there is a topology change. This exchange allows routers to automatically learn about new networks and also to find alternate paths if there is a link failure to a current network. Static routing over dynamic routing protocols requires less administrative overhead, the actual procedure is shown in Fig 1. However, the expense of using dynamic routing protocols is dedicating part of a router's resources for protocol operation, including CPU time and network link bandwidth. Despite the benefits of dynamic routing, static routing still has its place. There are times when static routing is more appropriate and other times when dynamic routing is the better choice.

IV. THEORETICAL ANALYSIS

In this section we analyze and briefly discuss the Tools and algorithms which we used to Emphatic import of conditional dynamic shortest path routing in Delay Tolerant Network. In our discussion, we shall first examine the algorithms and their work. In this, we implement a shortest path routing based algorithms. Based on these algorithms we find the conditional intermeeting time. We propose Conditional Shortest Path Routing (CSPR) protocol that route 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.

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 (Eu) and bidirectional (Eb) edges between the nodes. The neighbors of a node i are denoted with $N(i)$ and the edge sets are given as follows: 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). 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.

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 $n0$ to a node nd 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 $n0$ with $n1$ and $\Re_{n_0}(n_1|t)$ is the expected residual time for node $n0$ to meet with node $n1$ given that they have not met in the last t time units. $\Re_{n_0}(n_1|t)$ Can be computed 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 *τ* 1

$$\Re_i(j|t) = \frac{\sum_{k=1}^{d} f_i^k(j)}{|\{\tau_i^k(j) \geq t\}|} \text{ where,}$$
$$f_i^k(j) = \begin{cases} \tau_i^k(j) - t & \text{if } \tau_i^k(j) \geq t \\ 0 & \text{otherwise} \end{cases}$$

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 $\Re_i(j|t) = 0$. . We will next provide an example to show the benefit of CSP over SP. 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 their 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 *τD*(*E*|*B*) ≥ *τD*(*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 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 $G' = (V', E')$ where:

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

Note that the edges in *Eb* (in *G*) are made directional in *G'* and the edges in *Eu* 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 *τ* 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 $(i_S) \in V'$ 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 *τi*(*D*|*j*) and the algorithm is shown in Algorithm 1.

ALGORITHM 1
update (node m, time t)
1:if m is seen first time then
2: firstTimeAt[m] ← t
3: else
4: increment β_m by 1
5: lastTimeAt[m] ← t
6: end if
7: for each neighbor j € N and j ≠ m do
8: start a timer tmj
9: end for
10: for each neighbor j € N and j ≠ m do
11: for each timer tjm running do
12: S[j][m] += time on tjm
13: increment C[j][m] by 1
14: end for
15: delete all timers tjm
16: end for
17: for each neighbor i € N do
18: for each neighbor j € N and j ≠ i do
19: if S[j][i] ≠ 0 then
20: T_s(i|j) ← S[j][i] / C[j][i]
21: end if
22: end for
23: T_s(i) ← (lastTimeAt[i] − firstTimeAt[i]) / β_i
24: end for

V. MODULES IMPLEMENTED

Modules implemented in this paper as follows:

Networking Module: Client-server computing or networking is a distributed application architecture that partitions tasks or workloads between service providers (servers) and service requesters, called clients. Often clients and servers operate over a computer network on separate hardware. A server machine is a high-performance host that is running one or more server programs which share its resources with clients. A client also shares any of its resources; Clients therefore initiate communication sessions with servers which await (listen to) incoming requests.

Shortest Path Module: In multi-hop wireless networks, packets are transferred through routes that could be composed of multiple relay nodes between sources and destinations. In many multi-hop wireless networks, shortest path routing is often used for its simplicity and scalability, and this is closely approximated by straight line routing for large multi-hop wireless networks. Thus, in this paper, we will focus on straight line routing for delivering packets from sources to destinations.

Straight Line Routing Module: Both simulations and analysis show that the relay load over the network, imposed by straight line routing, depends on the model of the traffic pattern. Even if the system settings are identical and straight line routing is commonly adopted, the relay load induced by “random” traffic could be distributed differently over the network. This paradoxical result is a consequence of the famous Bertrand’s paradox. Thus, in contrast to traditional belief, there are many scenarios in which straight line routing itself can balance the load over the network, and in such cases explicit load-balanced routing may not help mitigate the relaying load.

Multi Hop Module: Analyze the load for a homogeneous multi-hop wireless network for the case of straight line routing in shortest path routing is frequently approximated to straight line routing in large multi-hop wireless networks. Since geographical and geometric attributes of nodes and routes affect the nodal load, we employ results from geometric probabilities to solve the problem. Based on our analytical results, we are able to show the precise relationship between the number of nodes and the load at each node, and the geographical distribution of the relaying load over the network for different scenarios. Interestingly, straight line routing itself can balance the relay load over the disk in certain cases.

VI. RESULTS

We conducted the simulation study from server to client in sending the file through conditional shortest path. We ran the conditional shortest path routing algorithm for it. Initial number of nodes that are selected for sending the file will be reduced after applying the algorithm, so that few nodes are selected as the shortest path, which are shown in Fig 2 and 3.

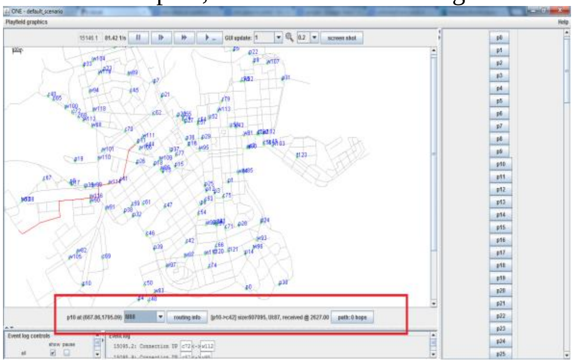

Fig 2 Results of shortest path

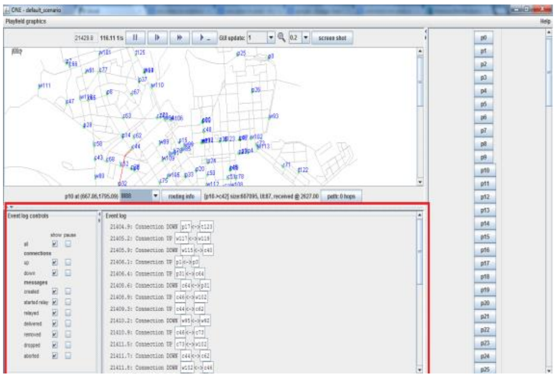

Fig 3 Results of Node connection

VII. COMPARATIVE STUDY

A Distributed adaptive fault- tolerant routing scheme is proposed for an injured hypercube in which each node is required to know only the condition of its own links. Despite its simplicity, this scheme is shown to be capable of routing messages successfully in an injured n-dimensional hypercube as long as the number of faulty components is less than n. Moreover, it is proved that this scheme routes messages via shortest paths with a rather high probability, and the expected length of a resulting path is very close so that of a shortest path. Since the assumption that the number of faulty components is less than n in an n-dimensional hypercube might limit the usefulness of the above scheme, a routing scheme based on depth-first search which works in the presence of an arbitrary number of faulty components is introduced. Due to the insufficient information on faulty components, however, the paths chosen by this scheme may not always be the shortest. In our system, defines the intermeeting time concept between nodes and introduces a new link metric called conditional intermeeting time.

It is the intermeeting time between two nodes given that one of the nodes has previously met a certain other node. This updated definition of intermeeting time is also more convenient for the context of message routing because the messages are received from a node and given to another node on the way towards the destination. Here, conditional intermeeting time represent the period over which the node holds the message. To show the benefits of the proposed metric, the project proposes conditional shortest path routing (CSPR) protocol in which average conditional intermeeting times are used as link costs rather than standard intermeeting times and the messages are routed over conditional shortest paths (CSP). By comparing 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.

VIII. CONCLUSIONS

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.

REFERENCES

[1] Delay tolerant networking research group, http://www.dtnrg.org.
[2] T. Spyropoulos, K. Psounis,C. S. Raghavendra, Efficient routing in intermittently connected mobile networks: The single-copy case, IEEE/ACM Transactions on Networking, vol. 16, no. 1, Feb. 2008.
[3] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, MaxProp: Routing for Vehicle-Based Disruption- Tolerant Networks, In Proc. IEEE Infocom, April 2006.
[4] A. Vahdat and D. Becker, Epidemic routing for partially connected ad hoc networks, Duke University, Tech. Rep. CS-200006, 2000.
[5] T. Spyropoulos, K. Psounis,C. S. Raghavendra, Efficient routing in intermittently connected mobile networks: The multi-copy case, IEEE/ACM Transactions on Networking, 2008.
[6] 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.
[7] S. Jain, K. Fall, and R. Patra, Routing in a delay tolerant network, in Proceedings of ACM SIGCOMM, Aug. 2004.
[8] T. Spyropoulos, K. Psounis,C. S. Raghavendra, Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks, ACM SIGCOMM Workshop, 2005.
[9] A. Lindgren, A. Doria, and O. Schelen, Probabilistic routing in intermittently connected networks, SIGMOBILE Mobile Computing and Communication Review, vol. 7, no. 3, 2003.
[10] 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.
[11] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In ACM SIGCOMM, 1994.
[12] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In IEEE WMCSA, 1999.
[13] E. M. Royer and C.-K. Toh. A review of current routing protocols for ad hoc mobile wireless networks. IEEE Personal Communications, 6:46–55, 1999
[14] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In ACM MobiCom, 2003.
[15] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In ACM MobiCom, 2004.
[16] 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.
[17] R. Handorean, C. Gill, and G.-C. Roman. Accommodating transient connectivity in ad hoc and mobile settings. Lecture Notes in Computer Science, 3001:305–322, March 2004.
[18] S. Srinivasa and S. Krishnamurthy, CREST: An Opportunistic Forwarding Protocol Based on Conditional Residual Time, in Proceedings of IEEE SECON, 2009.

[19] 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.
[20] C. Liu and J. Wu, Routing in a Cyclic Mobispace, In Proceedings of ACM Mobihoc, 2008.
[21] 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.
[22] S. Jain, K. Fall, and R. Patra, Routing in a delay tolerant network, in Proceedings of ACM SIGCOMM, Aug.

ABOUT AUTHORS:

V.PREMA TULASI working in Nalla Narasimha Reddy Education Society's Group of Institutions as Assistant Professor, Computer Science Department. Having 6 Years of experience in teaching and published papers in various journals.

K. **SATYANARAYANA** Working as Asst.prof, Dept of CSE in SPHOORTHY ENGINEERING COLLEGE. Had 7+ Years of experience in teaching, has done M.tech in Computer science from JNTU, Hyderabad. His area of interest is computer architecture - cache memory, pipelining, computer networking, data mining.

Mr. Revuri Varun, Post Graduated in Computer science and Engineering (M.Tech) , JNTUH , Hyderabad in 2011 and Graduated in Information Technology(B.Tech) from JNTUH, in 2007. He is working as an Assistant Professor in Department of Computer Science & Engineering in **Vaagdevi College Of Engineering**, Warangal Dist, AP, and India. He has 6 years of Teaching Experience. His Research Interests Include Network Security and Data Mining.

Mr. Varun Maramraju Post Graduated in Software Engineering (M.Tech), JNTUH, Hyderabad in 2012 and Graduated in Information Technology (B.Tech) from JNTUH, in 2008. He is working as an Assistant Professor in Department of Computer Science & Engineering in **Vaagdevi College Of Engineering**, Warangal Dist, AP, and India. He has 5 years of Teaching Experience. His Research Interests Include Network Security and Design and Analysis of Algorithms.