Characterization and Evaluation : Temporal Properties of Real and Synthetic Datasets for DTN

Hemal Shah¹, Yogeshwar Kosta², and Vikrant Patel¹

¹Computer Engineering, Ganpat University, Mehsana, Gujarat, India
²Faculty of Technology, MEFGI, Rajkot, Gujarat, India

Abstract - Node’s movements play a significant role in disseminating messages in the intermittently connected mobile ad-hoc network. In such networks scenarios traditional end-to-end paths do not exist; mobility creates opportunities for nodes to connect and communicate when they are encountered. A series of encountering opportunities spread a message among many nodes and eventually deliver to the destination. Further improvements to the performance of message delivery can come from exploiting temporal properties of intermittent networks. It is modeled as time varying graphs, where, moving nodes are considered as vertices and contact opportunity to other nodes as an edge. The paper discusses about characterization and design of the temporal algorithm. Then, evaluating temporal distance and temporal centrality of real and synthetic data sets. Such, characterization can help in accurately understanding dynamic behavior and taking appropriate routing decision.

Keywords: Temporal Graphs, Temporal Distance, Temporal Closeness Centrality, Temporal Path Length, Real Trace, Synthetic datasets.

1 Introduction

There are situations in mobile ad-hoc networks, where nodes are completely disconnected and may rely on relay nodes for contact opportunities to transfer the message. Such relay nodes create an opportunity for partial connectivity and carry the message until the next node or destination comes into contact[1]. In other networks, connectivity may exist, but only occasionally or intermittently. This intermittent connectivity is not failure or fault but, rather an integral part of dynamic networks. These networks are called Intermittently Connected Mobile Ad-hoc Networks (IC-MANET) also known as Delay Tolerant Networks (DTN)delay tolerant networks (DTNs)[2]. IC-MANET utilizes a Store-Carry-Forward[3] mechanism in which the intermediate node stores messages and forwards them to the nodes it encounters. In this manner, messages could be delivered to the destination hop-by-hop even if no stable end-to-end path exists. As network partition occurs frequently, if only one replica of the message[4] is kept, the message may reach the edge of partitioned network and be failed to be delivered at the destination. In order to increase the message delivery rate, each node can keep forwarded messages and copy them to other nodes it encounters. In this multiple-copy routing[5] manner, several replicas of the same message exist within the network.

In both single-copy routing and multiple-copy routing, the message delivery rate depends upon the node mobility, network connectivity and the intermediate node chosen strategy. Based on sufficient network connectivity, message delivery strategies should utilize node’s mobility characteristics to increase the message delivery rate and reduce network overhead. Studies looked[6] at analyzing static networks, i.e., networks that do not change over time. Given the collections of measurements related to real network traces, authors[7] are quickly starting to realize that connections are inherently time varying and exhibit more dimensionality than aggregate analysis can capture.

In time varying graph or temporal graph[6][8] vertices represent the node and opportunistic contact between nodes represent edge or links. These links are changing over the time and raising interesting questions:

- Are there any metrics [9][10][11] evolved or proposed by researchers relating to temporal graphs in IC-MANET?
- If available, can they be used to analyze real and synthetic data sets?
- Can the time varying behavior of mobile ad-hoc network be used for designing IC-MANET routing algorithms?

This has motivated to contribute towards defining the metrics related to temporal graph. Then, designing the temporal algorithm to evaluate metrics from real trace and synthetic data sets. Author’s contributions are:

1. Modeling intermittent network as time varying graphs.
2. Defining temporal measurement: temporal distance and temporal centrality.
3. Design and development of temporal characterization algorithm.
4. Evaluation of real and synthetic trace datasets for temporal properties.

Section 2 discusses IC-MANET as time varying graphs and defining temporal metrics. Section 3 discusses design temporal algorithm. Section 4 presents application real and synthetic dataset evaluation of temporal properties. Section 5 discusses about conclusion and future work.

2 Temporal Graph and Temporal Metrics

In IC-MANET, it is observed[10] that the connections are inherently varying over time and exhibit more dimensionality[12] than static network analysis can capture. Static graphs treat all links as appearing at the same time. It is unable to capture key temporal characteristics, and gives an overestimate of potential paths, connection pairs of nodes which cannot provide any information on the delay associated with an interaction. Thus, this does not retain temporal information and hence cannot capture the true duration or speed of dissemination. Instead, we now re-define a metric as the shortest temporal path length which gives an indication of the speed of message delivery from a source to destination. Before we can formalize this metric we first define the concepts of temporal paths. Following this, we then run through an example calculation of the temporal shortest path length and then define the algorithm that is used to compute the temporal distance.

2.1 Temporal Metrics

The shortest path length on static graphs returns the number of hops from a source node to destination node; this does not retain temporal information and hence cannot capture the true duration or speed of dissemination. Instead, we now re-define a metric as the shortest temporal path length which gives an indication of the speed of message delivery from a source to destination. Before we can formalize this metric we first define the concepts of temporal paths. Following this, we then run through an example calculation of the temporal shortest path length and then define the algorithm that is used to compute the temporal distance.

2.1.1 Temporal Path

For a given two nodes $i$ and $j$ temporal path defines as $p_{ij}(T_{min}, T_{max})$, where $T_{min}$ is the earliest contact time, at which $i$ can send a message to $j$, and $T_{max}$ is the latest contact time, at which $i$ can send a message to $j$. The shortest temporal path length from source $i$ to destination $j$ is defined as the minimum number of temporal hops, where $T_{min} \leq t \leq T_{max}$, necessary to transmit a message from $i$ to $j$. 

2.1.2 Temporal Distance

Given two nodes $i$ and $j$, the shortest temporal distance defines as $d_{ij}(T_{min}, T_{max})$ to be the shortest temporal path length, starting from time $T_{min}$, this can be thought as the number of time windows (or temporal hops) which takes for information to spread from a node $i$ to node $j$. The horizon $h$ indicates the maximum number of nodes within each window $G_{T}$ through which information can be exchanged, or in practical terms, the speed that a message travels. In the case of temporally disconnected node pairs $q,p$ i.e., information from $q$ never reaches $p$, then set the temporal distance $d_{pq} = \infty$.

2.1.3 Temporal Betweeness Centrality

Betweenness is commonly used to discover nodes that are critical for mediating information flow[13]. To identify these mediating nodes, the static betweenness centrality of a node $i$ is defined as the proportion of shortest paths between all pairs of nodes that pass through $i$. This proportion is important in that it gives a higher
weight to nodes which facilitate paths where there are no alternatives.

![Figure 2 Using Temporal Path length betweenness centrality](image)

To capture the notion of temporal betweenness it is important to take into account not only the proportion of shortest paths which pass through a node, but also the length of time for which a node along the shortest path retains a piece of information before forwarding it to the next node. For example, consider the 2-hop shortest temporal path from node 'P' to 'R', (P;Q;R) as shown in Figure 2.

![Figure 3 Betweenness centrality by consideration of time duration](image)

In terms of time, this path could be represented as (P:Q:Q:Q:Q:R) since a piece of information resides on node 'Q' for 4 time windows, and so we want to assign a higher value as removing this node will have a greater impact in disrupting the network as shown in Figure 3.

### 2.1.4 Temporal Closeness Centrality

The two nodes of a static graph are said to be close to each other if their geodesic distance is small. We can extend the definition of closeness to temporal graphs using the temporal shortest path length between nodes, which is a measure of how fast a source node can deliver a message to all the other nodes of the network.

Given the shortest temporal distance $d_{ij}(T_{\text{min}}, T_{\text{max}})$, temporal closeness centrality[7] can then be expressed as:

$$C_i^h = \frac{1}{W(N-1)} \sum_{v \in V} d_{ij}^h$$

So that, the nodes having, on average, shorter temporal distances to the other nodes are considered more central. Note that the subtraction from one is only required for a descending ranking.

### 3 Temporal Algorithm

Temporal distance $d_{ij}(T_{\text{min}}, T_{\text{max}})$, is computed in terms of number of time windows i.e.,

$$d_{ij}(T_{\text{min}}, T_{\text{max}}) = d_{ij}(T_{\text{min}}, T_{\text{max}}).$$

For each pair of i and j, algorithm computes $d_{ij}(T_{\text{min}}, T_{\text{max}})$ and then, takes the average of all values. This way temporal distance is computed in a number of time stamps. If average value multiplies with w, then result is the temporal distance in terms of time (in seconds). Eq. (1) gives the average temporal distance between $T_{\text{min}}$ and $T_{\text{max}}$:

$$L(T_{\text{min}}, T_{\text{max}}) = \frac{\omega}{N(N-1)} \sum_{ij} d_{ij}(T_{\text{min}}, T_{\text{max}})$$

### 3.1 Timewindow (w) Calculation

To understand the computation of Time window, refer Table 1 below showing calculation on dataset as an example, where each cell value represents the total contact time between a particular pair i,j divided by total the number of contact occurrences. For each node pair (i,j) compute a sum of all values. It returns the average meeting time per contact. The optimal value of time window is greater than average meeting time, because if time window <= average meeting time, then in most of the time windows, number of contact occurrence will be around one. That means, the information cannot be diffused efficiently into the network.

| Node ID | 1  | 2  | Total Contact Time | Total No. of Occurrences |
|---------|----|----|-------------------|--------------------------|
| 1       | 0/0| 480/2 | 480/2             |                          |
| 2       | 500/2 | 0/0 | 500/2             |                          |
| $\sum T_{ij}(T_{\text{min}}, T_{\text{max}})$ | 980 | $\sum N_{ij}$ | 245 Time Window size |

From above calculations it is established that, for effective information diffusion process into the network optimal time window should be greater than $\frac{\omega}{N(N-1)} \sum T_{ij}(T_{\text{min}}, T_{\text{max}})$ .E.g. In Figure 1(a) total number of time window = $(T_{\text{max}}-T_{\text{min}})/w = (900 - 0)/300 = 3$ timestamps, assuming time window size = 300.

Let’s find temporal distance $d_{ij}^w(t_{0}, t_{900})$ for the temporal graph shown in Figure 1(a). Here, $T_{\text{min}} = 0$ and $T_{\text{max}} = 900$. Time window size = 300. Thus, there are three time windows t1, t2 and t3.

### 3.2 Computation of Temporal Distance

Before starting calculation of temporal distance of each pair i,j, initialize number of empty lists equal to that of calculating number of time window. For each pair (i, j), i|j, start scanning timestamps from 1 to 3. For each timestamp, add occurred node id into the respective list of timestamp. Pair of node (i,j) occurs whenever there is a contact edge between node pair (i,j).
3.2.1 Preconditions
Pair of node \((i, j)\) occurs whenever there is a contact edge between node pair \((i, j)\).

Case 1:
If \(i = j\) then, return 0, in computing matrix below, temporal distance \((A, A) = (B, B) = (C, C) = (D, D) = 0\).

Case 2:
If both \(i\) and \(j\) occurs in same timestamp then return \((j\text{th timestamp number} - i\text{th timestamp number})\) or return \((0)\). In Figure 1(a), node A and node B occurs in same timestamp no. 1 , so the temporal distance between A and B is (B’s timestamp no. – A’s timestamp no.) = (1-1) = 0 timestamps.

Case 3:
If \(i\) occurs earlier than \(j\), then search occurrences of \(j\) in consecutive timestamps by using other occurred nodes in same timestamp in which \(i\) has occurred; for each pair \(i, j\) it may give more than one path in terms of required timestamp, in such a case select the shortest timestamp. In Figure 1(a), for temporal distance \((A,D)\), node A occurred in timestamp number 1 and node D occurred in timestamp number 3. Also, there is an intermediate node B which is common between node A and node D. So temporal distance \((A,D) = \) (node D’s timestamp number – node A’s timestamp number) = (3 – 1) = 2 timestamps.

Case 4:
If \(i\) occurs and \(j\) do not occur during a consecutive timestamp till \(T_{\text{max}}\), then the temporal path between a pair of \(i, j\) is not possible. So, return \(\infty\). Figure 1 (a), for a temporal distance \((D, E)\), node A occurred in timestamp number 1 and node D occurred in timestamp number 3. But there are no occurrences of node E also by using other intermediate occurrences of other nodes. So temporal distance \((D,E) = \infty\). In continuation of the example shown in Figure 1(a) and successive computation of temporal distance matrix as shown in Figure 2, the sum of non-negative values of matrix \(= 10\). Now, calculate the average temporal distance metric: \(300 \times (10 / (6) (5)) = 3000 / (30) = 100\). i.e., it takes average 100 seconds to reach from source ‘i’ to destination ‘j’.

\[
\begin{bmatrix}
0 & 0 & 1 & 1 & -1 & -1 \\
1 & 0 & 1 & 1 & -1 & -1 \\
1 & 1 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & -1 & -1 \\
-1 & 1 & 0 & 1 & 0 & 0 \\
-1 & 1 & 0 & 1 & 0 & 0
\end{bmatrix}
\]

Figure 2 computed temporal distance matrix values

3.3 Algorithm
1. Input source and target, \(T_{\text{min}}\) and \(T_{\text{max}}\) time window
Time Window Equation:
\[
w(T_{\text{window}}) = \frac{\sum t_i(t_{\text{min}}, t_{\text{max}})}{\sum N_{ij}} \tag{2}
\]
Where, \(\sum T_{ij} (t_{\text{min}}, t_{\text{max}}) = \) Total contact time between all pairs of nodes \(i, j\) and \(\sum N_{ij} = \) Total occurrences of all pairs of nodes \(i\) and \(j\).

2. Number of times frames = \(T_{\text{max}} - T_{\text{min}} / \) Time window.

3. Initialize number of empty list equal to number of time frames. Each list shows node ids whose contact occurred in a respective time frame.

4. Read the dataset and perform a lookup for node contact in different time frames and generate a distance matrix for each node. Per contact frame, fill up the array / list with node ids in contact.

5. Compute the temporal distance as:
   a. If source and target ids are in the same list, return (target time frame number – source time frame number) as temporal distance.
   b. Otherwise, look up source and target in different time frames. If the source time frame < target time frame then return (target time frame number – source time frame number) as temporal distance.
   c. In case repeated occurrence of the source, target sets \(T_{\text{min}}= \) last target occurred +1 timestamp and repeat steps a and b.

6. Take average values of all pairs (source, target) temporal distance.

7. Repeat steps 4,5,6 and 7 for all pairs(source, target) and generate matrix. Minus one (-1) indicates no edge between a pair of nodes in the matrix.

4 Application of Temporal Algorithm
First network topology is generated from large real data sets using python custom made script. Python provides a module called networkX , which helps to generate network topology according to the dataset. For evaluation, we have downloaded the INFOCOM’06, RollerNet real trace data from CRAWDAD and Random Way Point (RWP) generated using ONE simulator. Real and Synthetic data set information are presented in Table 2 (a) and Table 2 (b).

| Datasets | RollerNet | RWP_63 |
|----------|-----------|--------|
| Start Date | 2/2/2009 | NA |
| Duration | 0.12 days | 0.12 days |
| \(T_{\text{min}}-T_{\text{max}}\) | Day 1: 0-3096(51.6 min) | Day 1: 0-3096 |
| Number of Nodes | 63 | 63 |
| Contacts | 80824 | 576 |

Table 2 (a) RollerNet and RWP Dataset details
4.2 Temporal Distance

Temporal algorithm uses the value of $T_{\text{max}} - T_{\text{min}}$, number of connections and timestamps, time window size as input computed and presented in Table 3. Then, it evaluates the temporal distance and centrality metrics as below.

### Table 2 (b) INFOCOM’06 and RWP Dataset details

| Datasets | INFOCOM’06 | RWP 98 |
|----------|------------|--------|
| Start Date | 13/03/2005 | NA     |
| Duration | 4 days | 1 day |
| $T_{\text{min}} - T_{\text{max}}$ | Day 1: 61260 - 86400 (6.98 hours) Day 2: 86400 - 172800 (24 hours) Day 3: 172800 - 259200 (24 hours) Day 4: 259200 - 345600 (24 hours) | Day 1: 0-342915 |
| Number of Nodes | 98 | 98 |
| Contacts | 118875(of all four days) | 4412929 |

For any real trace or synthetic dataset, it is required to set the common format as shown in Table 3 in order to carry out time window calculations. The customized scripts are written taking care of converting datasets into desired format.

### Table 3 Common format for data sets

| Source Node ID | Destination Node ID | CONNECTION UP Time | OCCURRENCE COUNT | INTER CONTACT TIME |
|----------------|---------------------|--------------------|------------------|-------------------|
| 1              | 3                   | 51293              | 1                | 0                 |
| 1              | 3                   | 60603              | 2                | 9310              |
| 1              | 3                   | 62363              | 3                | 1760              |
| 1              | 3                   | 79649              | 4                | 17286             |

**4.3 Centrality Evaluation**

The existence of special nodes is vital due to their strong impact on message delivery. Degree centrality is measured as the number of direct ties that involves a given node. The node with highest degree centrality contacts the largest number of nodes in the network, so it is suitable to choose this node to act as a forwarder. Closeness centrality of nodes defines how long it takes information to spread from a given node to other nodes. The node with higher betweenness centrality has more chance to assist the communication on the link between two nodes. Table 5 shows the evaluated values of different centrality for real traces.

### Table 4 Temporal Distance evaluation for data sets

| Dataset Details | Number of Timestamps | Time window (w) (seconds) | Static Distance | Average Temporal Distance |
|-----------------|----------------------|---------------------------|-----------------|---------------------------|
| INFOCOM’06      | Day 1: 7 3240 | 1.56 3.97 |
| Day 2: 26 3240 | 1.23 14.26 |
| Day 3: 26 3240 | 1.3 12.8 |
| Day 4: 26 3240 | 1.3 11.75 |
| RollerNet       | Day 1: 207 15 | 1.22 297 |
| Day 44 71 | 3.64 0.46 |
| RWP 98          | Day 1: 2598 132 | 1.81 14.94 |

Temporal distance values presented in Table 4 gives us a better understanding of the network. Since, it can provide us an accurate measure of the delay of the information diffusion process that is not possible with traditional static metrics. In particular, since static shortest paths ignore time-order of contacts, it over-estimates the availability of contacts and therefore under-estimates the true shortest path.

### Table 5 Centrality evaluation for data sets

| Dataset Details | Diameter | Degree Centrality | Betweenness Centrality | Closeness Centrality |
|-----------------|----------|-------------------|------------------------|----------------------|
| INFOCOM’06      | Day 1: 4 (27, 0.81) | (85, 0.04) | (40, 0.83) |
| Day 2: 3 (56, 0.98) | (16, 0.02) | (56, 0.98) |
| Day 3: 3 (48, 0.95) | (51, 0.01) | (48, 0.95) |
| Day 4: 3 (44, 0.77) | (30, 0.05) | (44, 0.80) |
| RollerNet       | Day 1: 2 (51, 0.1) | (51, 0.0003) | (51, 0.1) |
| RWP 63          | Day 1: 6 (14, 0.29) | (14, 0.65) | (15, 0.42) |
| RWP 98          | Day 1: 2 (17, 0.99) | (17, 0.09) | (17, 0.99) |
It is observed that the values of temporal distance and centrality for the INFOCOM’06, RollerNet and RWP presented in section 4.2 and 4.3 are difficult to compare. Rather, our objective is to pop out time varying properties for efficient IC-MANET routing decision.

4.4 Observations

- Optimal time window size varies as per number of connections between nodes, number of nodes and total duration of $T_{\text{min}}$, $T_{\text{max}}$. Keeping the value < derived through script may result in overlooking connection and keeping too high will result in wastage of network resources.

- It found that synthetic data set values for temporal distance is poorer than real trees. This is due to its characteristics moving towards the center and random nature of the movement. Average temporal distance values of real trace analysis enables better routing decision and it is more accurate than static analysis.

- Diameter values can be used for evaluating maximum hops per time frame basis or on an average.

- Different centrality values help in identifying the important nodes. Such nodes can assist in the efficient information dissemination process.

- Referring the readings of RollerNet and RWP_63: It reveals that in RWP model the node movements are random and hence, the number of contacts and time stamps are less, resulting in lower average temporal distance value. It is seen that most of the time the nodes are moving around center due to which diameter, degree centrality, betweenness and closeness values are higher. These values clearly indicate the reasons (described above) behind not using the synthetic models for realistic scenarios. On the other hand, RollerNet data have comparatively higher contacts, and higher number of timestamps resulting better connectivity. Therefore, for efficient information dissemination these characteristics of dataset are being used by routing engines.

5 Conclusions

It reveals that the node mobility plays a vital role for the efficient diffusion of information in challenging environment. And while doing so one cannot ignore to understand the movement patterns and related properties such as time order, frequency, contact duration, inter contact time, etc. These dynamic properties of connection are first analyzed and understood by using time varying matrices: temporal distance, diameter and centrality. General framework has been to design carrying capability of evaluating temporal metrics from any synthetic and real trace data. Because such frameworks help in computing number of time frames and the size of time windows which in turn calculate temporal distance. These properties are very useful in designing the DTN routing protocol and understanding the dynamics of the network and thereby taking forwarding or replication decision.

6 Acknowledgment

We express our sincere gratitude to the management of Ganpat University – Mehsana and Marwadi Education Foundation - Rajkot; for providing us research opportunities and their wholehearted support for such activities. Finally, our acknowledgement cannot end without thanking the authors whose research papers helped us in making this research.

7 References

[1] W. Zhao, Y. Chen, M. Ammar, M. Corner, B. Levine, and E. Zegura, “Capacity Enhancement using Throwboxes in DTNs,” in 2006 IEEE International Conference on Mobile Ad Hoc and Sensor Systems, 2006, pp. 31–40.

[2] K. Fall, K. L. Scott, S. C. Burleigh, L. Torgerson, A. J. Hooke, H. S. Weiss, R. C. Durst, and V. Cerf, “Delay-Tolerant Networking Architecture,” IETF, vol. 54, no. 4838, 2007.

[3] B. D. Davison, “Store-and-Forward Performance in a DTN,” in 2006 IEEE 63rd Vehicular Technology Conference, 2006, vol. 1, pp. 187–191.

[4] W. Zhao, M. Ammar, and E. Zegura, “A message ferrying approach for data delivery in sparse mobile ad hoc networks,” in Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing - MobiHoc ’04, 2004.

[5] Amin Vahdat and David Becker, “Epidemic Routing for Partially Connected Ad Hoc Networks,” Tech. Rep. CS-200006, 2000.

[6] V. Kostakos, “Temporal graphs,” Physica A: Statistical Mechanics and its Applications, vol. 388, no. 6, pp. 1007–1023, Mar. 2009.

[7] N. Santoro, W. Quattrociocchi, P. Flocchini, A. Casteigts, and F. Amblard, “Time-Varying Graphs and Social Network Analysis: Temporal Indicators and Metrics,” Feb. 2011.
[8] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro, “Time-Varying Graphs and Dynamic Networks,” CoRR, Nov. 2010.

[9] M. E. J. Newman, “A measure of betweenness centrality based on random walks,” Social Networks, vol. 27, no. 1, pp. 39–54, 2005.

[10] J. Tang, M. Musolesi, C. Mascolo, and V. Latora, “Temporal distance metrics for social network analysis,” in Proceedings of the 2nd ACM workshop on Online social networks - WOSN ’09, 2009.

[11] A. Clauset and N. Eagle, “Persistence and periodicity in a dynamic proximity network,” DIMACS Workshop on Computational Methods for Dynamic Interaction Networks, Nov. 2012.

[12] J. Tang, M. Musolesi, C. Mascolo, and V. Latora, “Characterising temporal distance and reachability in mobile and online social networks,” ACM SIGCOMM Computer Communication Review, vol. 40, no. 1, Jan. 2010.

[13] J. Tang, M. Musolesi, C. Mascolo, V. Latora, and V. Nicosia, “Analysing information flows and key mediators through temporal centrality metrics,” in Proceedings of the 3rd Workshop on Social Network Systems - SNS ’10, 2010, pp. 1–6.