Time Message System for Delay Tolerant Networks

Vitor Guerra Rolla
Department of Informatics Engineering- CISUC
University of Coimbra
Pólo II - Pinhal de Marrocos
3030-290 - Portugal
Email: vitorgr@dei.uc.pt

Marilia Curado
Department of Informatics Engineering- CISUC
University of Coimbra
Pólo II - Pinhal de Marrocos
3030-290 - Portugal
Email: marilia@dei.uc.pt

***Abstract*—Communication in mobile ad hoc networks and delay tolerant networks seeks to address the technical routing issues of heterogeneous networks that may lack continuous network connectivity. This work proposes the Time Message System (TMS), a delay tolerant routing solution for wireless networks. The protocol predicts the distance in function of time between nodes according to the time of last meetings. TMS was designed for high node density IEEE 802.11 networks. Simulation results in such networks show that TMS can deliver the same number of messages earlier than a traditional delay tolerant routing solution well-known by the research community.**

***Index Terms*—routing, delay tolerant networks**

I. INTRODUCTION

Research about Delay Tolerant Networks (DTN) began with the need of networking technologies that can sustain the significant delays and packet corruption of space travel. Initially, DTN techniques were used specifically to Interplanetary Internet communications, however, with the widespread use of wireless technologies, the research community started to adapt some of its ideas to terrestrial wireless networks [1].

The growing numbers of IEEE 802.11 [2] wireless devices have made Mobile Ad-hoc Networks (MANET) a popular research topic since late 1990s. Such networks may operate by themselves or may be connected to the larger Internet. A MANET assumes that an end-to-end connection always exists from the origin to the destination. This assumption can be easily violated due to mobility in sparse unreliable networks. Thus, DTN seeks to address the technical issues in heterogeneous networks that may lack continuous network connectivity. When instantaneous end-to-end paths are difficult or impossible to establish, routing solutions must take to a ”store and forward” approach, where data is incrementally moved and stored throughout the network in hopes that it will eventually reach its destination [3]. Intermittent network connections may occur due to limits of wireless radio range, sparse mobile nodes, energy resources, attacks, or interferences. In order to cope with disconnections, the messages should be buffered for a long period, which means that nodes require extra buffer space to store messages that are waiting for future communication opportunities.

A basic classification for delay-tolerant routing solutions is whether or not the protocol creates replicas of messages. DTN routing solutions that never replicate a message are considered forwarding-based, whereas protocols that replicate messages are considered replication-based. This simple taxonomy was recently used in [4] to classify DTN routing protocols.

In forwarding-based DTN protocols, only a single copy of a message exists in storage in the network at any given time, therefore this approach wastes less network resources. Furthermore, when the destination receives the message, no other node has a copy, and thus there is no need to provide feedback to the network to indicate that outstanding copies can be deleted. However, forwarding-based approaches do not allow for sufficient message delivery rates in many DTN. An interesting study about the limitations of forwarding-based DTN routing solutions can be found in [5].

Replication-based protocols have obtained much attention from the research community due to their greater message delivery rates. Multiple message copies exist in the network, and only one must reach the destination. A common approach used to maximize the probability of a message being successfully delivered is to replicate many copies of the message in the hope that one will succeed in reaching its destination, an epidemic solution [6]. Replication-based routing solutions can be sub-classified in flooding-based and quota-based solutions [7]. In flooding-based solutions, if storage resources and mobility allow, it is possible for every node in the network to have a replica of the message. The quota-based solutions intentionally limit the number of replicas. Important issues in replication-based routing are waste of network resources, scalability, and congestion.

This work proposes the Time Message System (TMS), a DTN routing solution for IEEE 802.11 wireless networks with high node density and high mobility. The protocol was developed for networks without infrastructure. For instance, networks for emergency situations (e.g. earthquakes). The protocol predicts the distance between nodes according to the time of last meetings. The nodes that recently heard about the destination of a given message are more likely to deliver the message. The simulation results show that TMS can deliver almost the same number of messages that PROPHET [8] does, with a lower end-to-end delay in high node density network scenarios. TMS also has good delivery rates in less dense network scenarios, however PRoPHET delivers the messages faster when the network gets more sparse.

The next section presents the related work on DTN routing solutions. A basic overview of the PRoPHET protocol is given in section three, because it is the DTN routing solution used for comparison and evaluation. A detailed description about TMS forwarding process is given in section four. Simulations and performance comparison are presented in section five. Conclusions and future work are in the last section.

II. Related Work

Delay Tolerant Networks are characterized by their lack of connectivity, caused by limits of wireless radio range, sparsity of mobile nodes, energy resources, attacks, or interferences, which result in a lack of instantaneous end-to-end paths. This section of the paper presents well-known DTN routing solutions.

The Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET) [8] is a flooding-based DTN routing solution that relies on the calculation of delivery predictability to forward messages to the reliable node. The probability is used to decide if one node is reliable to forward a message to. A node that is often encountered has higher delivery predictability than the others. If two nodes do not encounter each other during an interval, they are less likely to be good forwarders of messages to each other, thus the delivery predictability values must be reduced. A basic overview of PRoPHET is given later in this section.

The MaxProp protocol [3] performs routing by considering the priority of packets to be transmitted, and the priority of packets to be dropped. It has a flooding-based nature, in such a way that whenever two nodes meet, all messages not held by a node will attempt to be replicated and transferred to the other one. The intelligence of MaxProp comes from determining which messages should be transmitted first and which messages should be dropped first. In practice, messages that are ranked with highest priority are the first to be transmitted during a transfer opportunity. Messages ranked with lowest priority are the first to be deleted to make room for an incoming message.

Resource Allocation Protocol for Intentional DTN (RAPID) [4] formulates the routing problem as a resource allocation problem. RAPID is flooding-based. The authors show that the DTN routing problem is NP-hard using a polinomial-time reduction from the edge-disjoint path problem for a directed acyclic graph [9]. RAPID is executed when two nodes are within range and have discovered one another. The protocol arranges the messages in order to choose a feasible schedule for transfers. It also assumes constraints on both storage capacity and available bandwidth. The key question solved by RAPID is: Given limited bandwidth, how can the messages be replicated in the network so as to optimize a specified routing metric? The protocol was deployed in a real vehicular network and simulated in a custom event-driven simulator.

Spray and Wait [10] is a quota-based DTN solution that attempts to limit the number of possible replicas of a given message. The protocol achieves resource efficiency by setting a strict upper bound on the number of copies per message allowed in the network. When a new message is created in the system, a number L is attached to that message indicating the maximum allowable copies of the message in the network. During the spray phase, the source of the message is responsible for ”spraying”, or delivering, one copy to L distinct DTN nodes. When a relay receives the copy, it enters the waiting phase, where the relay simply holds that particular message until the destination is encountered directly.

Encounter-based routing (EBR) [11] argues that nodes with more encounters are more likely to successfully pass data along to the final destination than the nodes who only infrequently meet others. Every node running EBR is responsible for maintaining two pieces of information: an encounter value and a current window counter for the calculation of past rate of encounter average. EBR is quota-based. A similar approach is used in [12], however, the authors explore the idea that more encounters between two nodes means the more these nodes are expected to meet. Consequently, less is the benefit that they carry the same messages.

Bubble Rap [13] is a DTN routing protocol focused on two specific aspects of society: community and centrality. The Bubble Rap is forwarding-based and has the following assumptions: each node has labels that inform other nodes of its community’s affiliation; and, each node has a global centrality across the whole system, and also a local centrality within its communities. Such centralities are calculated over a social graph, where there is an edge between two nodes if there has been at least one contact between them at any time in the past. The routing decision is to forward messages to nodes which are more popular than the current node.

The Geographic DTN Navigation (GEODTN+Nav) routing solution [14] was proposed for urban vehicular environments. It is a geographical routing forwarding-based DTN protocol. When in DTN mode, the solution uses data-mule nodes to guarantee message delivery in segmented network partitions. The paper assumes that each node (vehicle) has a virtual interface navigation (which interacts with the GPS device) to provide two types of information: the detailed path, the destination or just the node direction and the confidence about the path. Low confidence means great random mobility (e.g. taxi node), while high confidence means more capacity of mobility prediction (for example: bus or train node). When a message is generated in the system, it is forwarded in greedy mode. The routing solution only gets into DTN mode when the packet has taken a high number of nodes in the perimeter mode. When in DTN mode, the node responsible for storing the messages waits for a neighbour node presenting a new mobility pattern that shall bring the packet near to its destination. Thus, these nodes act as a data-mule in the network. The neighbour selection to carry the packet is based on the level of confidence. The protocol was evaluated with realistic vehicular mobility traces.

Context-aware adaptative routing (CAR) [15] makes use of Kalman Filter [16] prediction to choose the best carrier for the message. This method of discrete processing provides optimal estimates of the current state of a dynamic system described by a state vector, and is used to calculate a more realistic prediction of the evolution of the context of a node. The authors define context as the set of attributes that describe the aspects of a system that can be used to drive the process of message delivery, namely: change degree of connectivity (number of connections and disconnections that a node experienced over the last *T* seconds), the future node colocation, battery level, memory availability or group membership. CAR focuses on the first two of these contexts, composing them into a single utility value, which represents how good a node is for delivering messages for a specific destination. This context information measures relative mobility and the probability that a node will encounter other nodes.

The Hybrid DTN-MANET (HYMAD) routing solution [17] is composed of an intra-group proactive distance vector protocol for MANET routing and an inter-group routing protocol which is based on Spray and Wait for DTN routing. The intra-group protocol runs a distributed algorithm to segment the network in partitions accordingly to a pre-defined group diameter. Border nodes are responsible for the inter-group routing. When a border node learns about DTN packets inside its groups, they have the following options: if one of the neighbour groups is the destination of the packet, then forwards it to the neighbour group; or, if the neighbours groups are not the destination of the packet, the algorithm splits the total number of packet copies permitted between all border nodes and forwards it to the adjacent groups. Since HYMAD makes use of Spray-and-Wait it is also a quota-based routing solution.

The Delay-tolerant Dynamic MANET On-demand (DT-DYMO) [18] routing solution combines the Dynamic MANET On-demand Routing protocol (DYMO) and PRoPHET. DT-DYMO works as follows: if a route request does not reach its destination, the origin node specifies a minimum delivery probability for the potential DTN nodes. All nodes that exceed this delivery probability answer with a Route Reply, offering to store the packet. The protocol can operate either in DTN flooding-based (multiples copies of the packet in the network) or in DTN forwarding-based mode (only one copy of the packet in the network).

Since computer network research on DTN is vast, the academic community addresses the issue from different perspectives. PRoPHET, Spray-and-wait and EBR were designed without regard to any additional information about the network scenario; they are generic approaches that can be used in any DTN scenario. RAPID, MaxProp and GEODTN+Nav were designed for IEEE 802.11 vehicular networks; GEODTN+Nav makes use of the GPS network. Bublle Rap proposes the use of Complex Network Analysis (CNA) [19] to formulate contact prediction social-based DTN routing for bluetooth networks. CAR, HYMAD and DT-DYMO are IEEE 802.11 MANET routing solutions that incorporate DTN mechanisms to be robust from an end-to-end perspective, and at the same time, be resilient enough to tolerate intermittent connectivity.

III. PRoPHET Overview

PRoPHET is a flooding-based DTN routing solution that relies on the calculation of delivery predictability to forward messages to the reliable node. The probability is used to decide if one node is reliable to forward a message to. A node that is often encountered has higher delivery predictability than the others. If two nodes do not encounter each other during an interval, they are less likely to be good forwarders of messages to each other, thus the delivery predictability values must be reduced.

In each node, an adaptive algorithm is used to determine the delivery predictabilities *P(o,d)*. The node *o* stores delivery predictabilities *P(o,d)* for each known destination on *d*. If the *od* pair has not encountered themselves the predictability value *P(o,d)* is assumed to be zero. The delivery predictabilities used by each node are recalculated at each opportunistic encounter according to the following terms:

- When node *o* encounters node *d*, the predictability for *P(o,d)* is increased. This calculation is shown in Eq. 1, where $P_{init} \in [0,1]$ is an initialization constant.

$$P_{(o,d)} \leftarrow P_{(o,d)old} + (1 - P_{(o,d)old}) * P_{init} \tag{1}$$

- The predictabilities for all destinations other than *d* are aged; this process is based on a constant aging and the number of time units (sec) that have elapsed since the last time the metric was aged. If a pair of nodes does not encounter each other in a while, they are less likely to be good forwarders of messages to each other, thus the delivery predictability values must age. The aging calculation is shown in Eq. 2, where $\gamma \in [0,1)$ is the aging constant, and κ is the number of time units that have elapsed since the last time the metric was aged. The time unit shall be based on the application and the expected delays in the targeted network.

$$P_{(o,d)} \leftarrow P_{(o,d)old} * \gamma^{\kappa} \tag{2}$$

- Predictabilities between *o* and *d* are exchanged and updated using its transitive property. This property is based on the observation that if node *o* frequently encounters node *d*, and node *d* frequently encounters node *x*, node *x* probably is a good node to forward messages destined for node *o*. Eq. 3 shows how this transitivity affects the delivery predictability, where $\beta \in [0,1]$ is a scaling constant that decides how large impact the transitivity should have on the delivery predictability.

$$P_{(o,d)} \leftarrow P_{(o,d)old} + (1 - P_{(o,d)old}) * P_{(o,d)} * P_{(d,x)} * \beta \tag{3}$$

A simple example of a routing strategy for PRoPHET protocol is: when two nodes meet, a message is transferred to the other node if the delivery predictability of the destination of the message is higher at the other node.

IV. TMS

TMS was designed for high node density IEEE 802.11 networks, because of the proliferation of such technology

(a) Distance-table of node *x* before the encounter with node *y*.

Known Node	**Last time heard about node**	**Distance**
a	13:34:00hrs	2s
k	13:35:17hrs	4s

(b) Distance-table of node *x* after the encounter with node *y*.

Known Node	**Last time heard about node**	**Distance**
a	13:36:01hrs	5s
k	13:35:17hrs	4s
y	13:36:01hrs	1s
i	13:36:01hrs	29s
j	13:36:01hrs	76s

TABLE I

DISTANCE-TABLE OF NODE *x* BEFORE AND AFTER THE ENCOUNTER WITH NODE *y*.

(a) Distance-table of node *y* before the encounter with node *x*.

Known Node	**Last time heard about node**	**Distance**
i	13:35:42hrs	10s
j	13:35:07hrs	22s
a	13:35:58hrs	2s

(b) Distance-table of node *y* after the encounter with node *x*.

Known Node	**Last time heard about node**	**Distance**
i	13:35:42hrs	10s
j	13:35:07hrs	22s
a	13:35:58hrs	2s
x	13:36:01hrs	1s
k	13:36:01hrs	48s

TABLE II

DISTANCE-TABLE OF NODE *y* BEFORE AND AFTER THE ENCOUNTER WITH NODE *x*.

in densely populated areas. The nodes that recently heard about the destination of a given message are more likely to deliver the message. While PRoPHET transitively computes delivery predictabilities to forward messages to the reliable node, TMS transitively computes the distance in function of time between nodes meetings. This is perfect to deal with short disconnections present in high node density wireless networks, because the distance between nodes can be used in the forwarding strategy. The system has the following premisses:

- Each node has a buffer to store DTN data messages. Each DTN data message has a time-stamp. When the buffer is full, the oldest DTN data message is discarded.
- Each node has a distance-table that indicates the distance in function of time from the source node to all known nodes. The distance-table has three entries: known node address, last time heard about node, node distance at that particular time. Distance-table examples can be verified in tables I and II.
- Each node regularly broadcasts a control message with two pieces of information: its distance-table and the time-stamp the control message was created.

A. Distance-Table Algorithm

When a node *x* receives a control message from node *y*, it uses the control message time-stamp to update its temporal distance *D(x,y)*. This calculation is shown in Eq. 4, where, t_{now} is the time when node *x* process the control message and t_{cmts} is the control message time-stamp:

$$D_{(x,y)} \leftarrow t_{now} - t_{cmts} \tag{4}$$

Then, *x* compares the received distance-table from node *y* with its own distance-table:

- Any distinct node known by node *y* and unknown by node *x* is created in *x*'s distance-table. This update is shown in Eq. 5 and Eq. 6, where $t_{last}(x,t)$ is the last (updated) time that node *x* heard about the distinct node *z*.

$$t_{last}(x,z) \leftarrow t_{now} \tag{5}$$

$$D_{(x,z)} \leftarrow t_{now} - t_{last}(y,z) + D_{(y,z)} \tag{6}$$

- Distinct nodes known by node *y* that are also known by node *x*, shall be updated in *x*'s distance-table if and only if, Eq. 7:

$$t_{now} - t_{last}(x,z) + D_{(x,z)} > t_{now} - t_{last}(y,z) + D_{(y,z)} \tag{7}$$

A detailed description about TMS distance-table update process is given in the next subsection.

B. Distance-Table Update Description

For simplicity, assume an encounter of nodes *x* and *y* where only one control message was sent by each node at time 13:36:00hrs, and respectively received at time 13:36:01hrs. Table I represents the distance-table of node *x* before (table I-(a)) and after (table I-(b)) the encounter with node *y*. Consequently, table II represents the distance-table of node *y* before (table II-(a)) and after (table II-(b)) the encounter with node *x*.

From table I-(a) it can be noticed that node *x* already knows about nodes *a* and *k* before the encounter; at time 13:34:00hrs it was 2 seconds of distance from node *a*, and at time 13:35:17hrs it was 4 seconds to node *k*. From table II-(a) it can be noticed that node *y* already knows about nodes *i* and *j*, and also about node *a* before the encounter; at time 13:35:42hrs it was 10 seconds of distance from node *i*, at time 13:35:07hrs it was 22 seconds of node *j*, and at time 13:35:58hrs it was 2 seconds of node *a*.

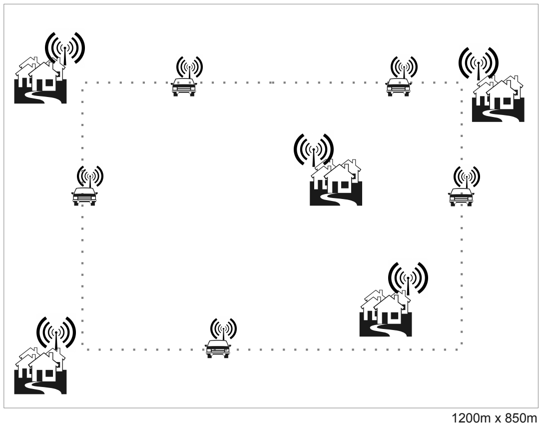

Fig. 1. Scenario 2 setup.

Upon receiving the control message from node *y* at 13:36:01hrs, the node *x* is able to calculate its directly temporal distance from the node *y* using the control message time-stamp. Please check the third entry of table I-(b). After that, the node *x* uses the distance-table broadcasted by node *y* in his control message to update each entry in its own distance-table accordingly:

- Known nodes by *y* that are unknown by *x*, which is the case of nodes *i* and *j*, are created in *x*'s distance-table. Please check the fourth and fifth entries of table I-(b). Since this is the first time that node *x* heard about those nodes, it is easy to calculate the distances:
  - For *i*: 13:36:01hrs (time that node *x* received the control message from *y*) - 13:35:42hrs (last time that node *y* heard about node *i*) + 10 seconds (the distance from *y* to *i*) = 29 seconds.
  - For *j*: 13:36:01hrs (time that node *x* received the control message from *y*) - 13:35:07hrs (last time that node *y* heard about node *j*) + 22 seconds (the distance from *y* to *j*) = 76 seconds.
- Known nodes by *y* that are also known by *x*, which is the case of node *a*, shall be updated in *x*'s distance-table if and only if:
  - For *a*: 13:35:58hrs (last time that node *y* heard about node *a*) - 2 seconds (distance of node *y* to node *a*) $>$13:34:00hrs (last time that node *x* heard about node *a*) - 2 seconds (distance of node *x* to node *a*).
  - Please check the first entry of table I-(b). It is easy to calculate the new distance from node *x* to node *a*: 13:36:01hrs (time that node *x* received the control message from *y*) - 13:35:58hrs (last time that node *y* heard about node *a*) + 2 seconds (the ditance from *y* to *a*) = 5 seconds.

Upon receiving the control message from node *y* at 13:36:01hrs, the node *x* follows the same procedure described above.

Simulation Parameters	
General	
Simulation Time	4000s
DTN data message size	50 bytes
DTN buffer size	2500 bytes
Playground size	1200m x 850m
Scenario 1	
N^{o}of pedestrian nodes	20
Mobility Model	Random waypoint
Speed (min/max)	1-2 m/s (pedestrian)
Pause time	0s
Scenario 2	
N^{o}of pedestrian nodes	20
Pedestrian mobility model	Random waypoint
Pedestrian speed (min/max)	1-2 m/s
Pause time	0s
N^{o}of car nodes	5
Car mobility model	Rectangle mobility
Car speed (min/max)	6-11 m/s
N^{o}of POI	5
TMS	
Control msg period	3 s
PRoPHET	
Init. predictability (P_{init})	0.75
Ageing (γ)	0.7
Predic. scaling factor (β)	0.25
Hello Interval	3s

TABLE III
SIMULATION PARAMETERS.

C. Forwarding Strategy

Whenever a given node updates its temporal distance from any other node on the network (by receiving a control message). This node checks in its buffer if there are DTN data messages for the destination whose the distance has been updated. A copy of each message for this destination is forwarded to the node that sent the control message. Using the same previous example, at 13:36:01hrs the node *x* checks in its buffer if there are messages to nodes *y*, *i*, *j* and *a*. If there are DTN data messages for those nodes, *x* forwards a copy of such messages to *y*, because it was the node that sent the control message.

The core idea of TMS is to forward DTN data messages to nodes that present short distances in function of time to the message's final destination. The protocol was designed to networks that present high node density with intermittent connectivity. It is a generic approach that can operate in forwarding-based (only one copy of each message exist in the system) and replication-based (multiple copies of the message in the system) DTN modes.

V. SIMULATION

The simulations were performed using the OMNeT++ network simulator version 4.1 with the INETMANET framework

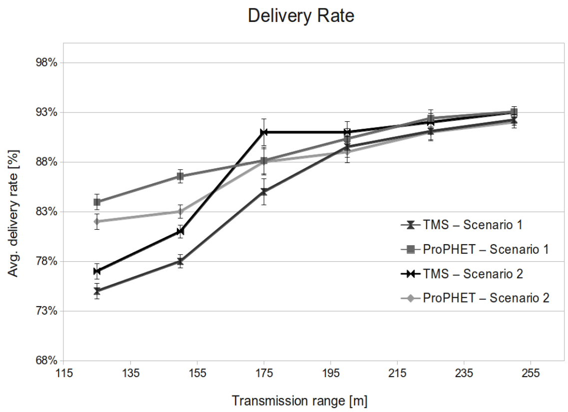

Fig. 2. Delivery rate with different network densities.

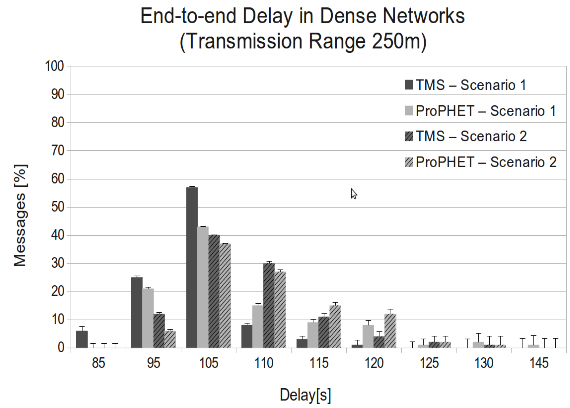

Fig. 3. End-to-end histogram in dense mobile networks.

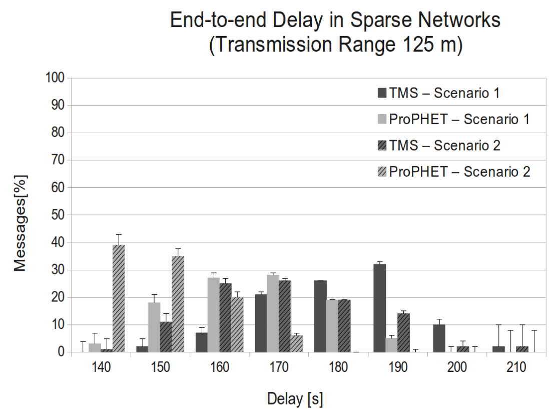

Fig. 4. End-to-end histogram in sparse mobile networks.

[20]. TMS was compared to PRoPHET, because it is well-known by the research community and can achieve good delivery rates in heterogeneous network scenarios. PRoPHET reference implementation is maintained by the Internet Research Task Force (IRTF[1]) DTN Research Group. Both protocols were implemented as network layer modules on the INETMANET framework.

A. Setup

The IEEE 802.11 Layer in ad-hoc mode was used with free space propagation model on the physical layer. The application layer generates DTN data messages to random destination nodes every 30 seconds after an initial phase of 10 min, for properly PRoPHET delivery predictabilities setup. The playground size used was 1200m x 850m. All nodes have synchronized clocks. Different transmission ranges were applied in order to simulate sparse and dense networks. The data was collected over 10 simulation runs for each network density.

TMS and PRoPHET were executed in two scenarios:

1) Scenario 1: all nodes were placed randomly and start moving continuously with pedestrian speed according to the random waypoint model without pause time.
2) Scenario 2: pedestrian nodes were placed randomly and start moving continuously according to the random waypoint model without pause time. Car nodes move in rectangular mobility and the points of interests (POI) do not move. Figure 1.

The parameters used in the simulations are given in table III. Simulation scenarios were based on [17] and [18].

B. Results

Figure 2 shows the delivery rate for different transmission ranges (network densities) for both proposed scenarios. As expected, in sparse networks (transmission range <150m) TMS delivers fewer messages than PRoPHET. Sparse networks lead to long distances. By design, long distance means less knowledge of the network neighbors in the TMS protocol. With increasing density, TMS achieves delivery rates close to PRoPHET in scenario 1. Though, in scenario 2 TMS has better delivery rates than PRoPHET. These results show the effectiveness of TMS routing procedure on networks that have a high node density with intermittent connectivity.

PRoPHET's delivery probabilities do not always reflect the shortest distance to destination, thus its message delivery takes longer than TMS in dense networks. This is visible in the histograms for end-to-end delay in figure 3. In dense networks, TMS proves the advantage of distance in function of time routing in both scenarios.

- Scenario 1: over 90 per cent of the messages are delivered within the first 105 sec. PRoPHET delivers the majority of the messages slower, 66 per cent within the same simulation time.
- Scenario 2: over 82 per cent of the messages are delivered within the first 110 sec. PRoPHET delivers the majority of the messages slower, 72 per cent within the same simulation time.

[1]IRTF focuses on longer term research issues related to the Internet while the parallel organization, the Internet Engineering Task Force (IETF), focuses on the shorter term issues of engineering and standards making.

However, PRoPHET has better results than TMS in sparse networks, as shown in figure 4.

- Scenario 1: PRoPHET delivers almost all messages within 180 sec of simulation time, TMS only delivers 50 per cent of the messages in the same simulation time.
- Scenario 2: PRoPHET delivers almost all messages within 160 sec of simulation time, TMS only delivers 38 per cent of the messages in the same simulation time.

TMS was developed for networks without infrastructure. For instance, networks for emergency situations. These results prove that TMS can achieve good delivery rates and reduce delay in high node density IEEE 802.11 networks with intermittent connectivity.

VI. Conclusions and Future Work

TMS predicts the distance between nodes according to the time of last meetings. The nodes that recently heard about the destination of a given message are more likely to deliver the message. The most important conclusion of this work until now is that even with simplistic routing procedures TMS is a feasible DTN routing solution for networks with high node density and high mobility. Though, the simulation results show that TMS can deliver almost the same number of messages that PROPHET does, within a lower end-to-end delay in such networks.

TMS project future tasks is provide feedback to the network to indicate that outstanding copies of delivered messages can be deleted and enhance the buffer management for a better selection of messages to be tranferred in a given encounter and to be dropped from the buffer.

Acknowledgement: Fundação para Ciência e Tecnologia (FCT) User Centric Routing (UCR) Project (PTDC/EEA-TEL/103637/2008).

References

[1] K. Fall, "A delay-tolerant network architecture for challenged internets," *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications - SIGCOMM '03*, p. 27, 2003.

[2] "Ieee standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pp. C1 –1184, 12 2007.

[3] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks," *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, vol. 00, no. c, pp. 1–11, 2006.

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

[5] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Efficient Routing in Intermittently Connected Mobile Networks: The Single-Copy Case," *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, pp. 63–76, Feb. 2008.

[6] A. Vahdat, D. Becker, and Others, "Epidemic routing for partially connected ad hoc networks," Tech. Rep., 2000.

[7] Q. Yuan and I. Cardei, "An Effiicient Prediction-based Routing in Disruption-Tolerant Networks," *Parallel and Distributed Systems, IEEE*, pp. 1–31.

[8] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 7, no. 3, p. 19, Jul. 2003.

[9] R. Aharoni and E. Berger, "Mengers theorem for infinite graphs," *Inventiones mathematicae*, vol. 176, no. 1, pp. 1–62, Dec. 2008.

[10] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*. ACM, 2005, pp. 252–259.

[11] S. C. Nelson, M. Bakht, and R. Kravets, "Encounter-Based Routing in DTNs," *IEEE INFOCOM 2009 - The 28th Conference on Computer Communications*, pp. 846–854, Apr. 2009.

[12] T. Abdelkader, K. Naik, and A. Nayak, "A socially-based routing protocol for delay tolerant networks," *GLOBECOM IEEE Global Telecommunications Conference*, 2010.

[13] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: social-based forwarding in delay tolerant networks," *IEEE Transactions on Mobile Computing*, 2010.

[14] P.-C. Cheng, K. C. Lee, M. Gerla, and J. Härri, "GeoDTN+Nav: Geographic DTN Routing with Navigator Prediction for Urban Vehicular Environments," *Mobile Networks and Applications*, vol. 15, no. 1, pp. 61–82, Jun. 2009.

[15] M. Musolesi, "Car: Context-aware adaptive routing for delay-tolerant mobile networks," *IEEE Transactions on Mobile Computing*, pp. 1–14, 2008.

[16] J. Hamilton, *Time Series Analysis*. Princeton University Press, 1994.

[17] J. Whitbeck and V. Conan, "HYMAD: Hybrid DTN-MANET routing for dense and highly dynamic wireless networks," *Computer Communications*, vol. 33, no. 13, pp. 1483–1492, 2010.

[18] C. Kretschmer, S. Ruhrup, and C. Schindelhauer, "DT-DYMO: Delay-Tolerant Dynamic MANET On-demand Routing," *2009 29th IEEE International Conference on Distributed Computing Systems Workshops*, pp. 493–498, Jun. 2009.

[19] M. E. J. Newman, "The Structure and Function of Complex Networks," *SIAM Review*, vol. 45, no. 2, p. 167, 2003.

[20] Omnet++ simulator. [Online]. Available: www.omnetpp.org