

NECTAR: A DTN Routing Protocol Based on Neighborhood Contact History

Etienne C. R. de Oliveira and Célio V. N. de Albuquerque
Instituto de Computação - Universidade Federal Fluminense
{eoliveira,celio}@ic.uff.br

ABSTRACT

There are a number of scenarios where connectivity is intermittent, and a given destination may not be reachable at the moment a message is sent. Networks with these characteristics are known as Delay and Disruption Tolerant Networks (DTN). The NECTAR protocol proposed in this article is based on the contacts history in order to create a Neighborhood Index and then determine the most appropriated route for DTNs. Simulations performed with real data retrieved from mobile and wireless environments at Dartmouth College in scenarios where the occurrence of highly-partitioned networks is frequent, and with the presence of resource constrained nodes show that NECTAR is able to deliver more messages than Epidemic and PROPHEt protocols with lower consumption of network resources.

Categories and Subject Descriptors

C.2.2 [Network Protocols]: Routing protocols

General Terms

Routing protocols

Keywords

DTN, Routing Protocol, Neighborhood

1. INTRODUCTION

With the real possibility of ubiquitous communication, wireless mobile devices may establish communication at any moment and in any place, regardless of the existence or not of an end-to-end path between source and destination. In such environments, the occurrence of faults or the mobility pattern of some devices can cause network partitions, creating highly-partitioned networks.

As the vast majority of routing protocols assumes that there must exist an end-to-end path between source and destination, some networks that suffer from occasional or frequent partitioning may have significant levels of inefficiency

or, during specific intervals, a total incapacity of communication. The lack of an end-to-end path between source and destination is handled by MANET routing protocols as a temporary and unexpected disconnection. However, there are specific scenarios where the end-to-end path between source and destination may not exist for hours, days or even weeks. In some cases, it may never exist, as described in [2, 3, 5], in generic scenarios such as rural communications, battlefields communications, and communications in disasters environments. Those kinds of networks are known as Delay and Disruption Tolerant Networks (DTN).

Some routing protocols for MANETs and DTNs are validated on artificial scenarios with random mobility models. Lindgren et al. [11] asserted that artificial scenarios with random mobility models are unable to reflect, faithfully, real movements patterns. Real users do not move in a random way. If a person visits one specific location several times, we can predict that, based on the previous behavior, the person will visit that location again. In real stochastic scenarios, a node movement does not occur in a totally random way, i.e., each node moves toward a certain destination that may be visited many times. During movement, there is a greater likelihood that nodes meet again some of their past neighbors. The neighbors of a node X may move toward a certain destination, establish contact with others nodes, and may spread the information that there was a known route to node X . Nodes that learned this route may move toward others destinations, and may spread the route to node X to a new set of neighbors, and so on. Based on this heuristic, we expect that messages sent in a controlled manner to nodes that once have been in the neighborhood of a destination have their delivery probability increased. The spread of the neighborhood information enables a more accurate understanding of the network topology, which, inherently, aids the routing task. Students with PDAs on a university *campus*, lecturers with laptops in a conference, or children with the OLPC laptop [12] moving toward the school and returning to their homes are examples of real scenarios.

The routing protocol proposed in this article uses the concept of neighborhood to route messages in stochastic scenarios. Traces of real movements, extracted from CRAWDAD [7] at Dartmouth College, were used in order to evaluate the proposed NECTAR Routing Protocol. In this work, NECTAR is compared to two protocols, Epidemic [17] and PROPHEt [11] due to their distinct characteristics. The delivery rate of Epidemic is usually considered to be optimal by many previous works, such as [8, 11, 13, 14]. However, this protocol does not perform well in highly constrained

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC'09 March 8-12, 2009, Honolulu, Hawaii, U.S.A.

Copyright 2009 ACM 978-1-60558-166-8/09/03 ...\$5.00.

scenarios, due to buffer overflow and network congestion. The PROPHET protocol presents similarities with our proposal and is one of the few routing protocols for DTNs which has an Internet-Draft [10]. Nevertheless, it seems to update the routing table in an excessively slow fashion. The results presented in this work corroborate these statements. The NECTAR protocol intends to perform well in scenarios with high resources constrained and to deal with very high traffic and frequent node movements.

This paper is organized as follows: Sections 2 and 3 present, briefly, the Epidemic and PROPHET routing protocols; Section 4 describes the NECTAR protocol; Section 5 presents the scenario and methodology used in simulations, and evaluates the results; Section 6 presents some existing related work; and Section 7 summarizes our conclusions and describes future work.

2. EPIDEMIC ROUTING

The Epidemic Routing protocol [17] is one of the earliest proposals for routing in DTNs. The basic operation is extremely simple: data source nodes and intermediate nodes flood messages to all its neighbors to mitigate the effects of a single path failure, so that, eventually, the message may arrive at the destination node. Messages are quickly distributed through the neighborhood, but significant resources from network and nodes may be wasted in this process. This approach can achieve high delivery ratios, and does not need any previous knowledge of the network.

A structure called summary vector indicates which messages are stored. To avoid redundant message transfer during an anti-entropy session, each node maintains a list of nodes that it has established contact recently. During the anti-entropy session, the two nodes exchange their summary vectors, identify any previously unseen messages, and request copies. In order to control the dissemination process, messages' header contains a field called hop count, whose function is similar to the TTL field (time to live) of the IP protocol. Messages with hop count equal to one will only be forwarded to the destination node.

3. PROPHET

The PROPHET protocol [11] estimates the delivery probability based on the history of encounters. A metric called Delivery Predictability, $P_{(a,b)} \in [0, 1]$, is calculated for every node a for each known destination b . Suppose that a node a has a message m for the destination d . When a contact occurs between a pair of nodes a and b , node a forwards the message m to node b only if b has a greater Delivery Predictability to the destination d , that is, $P_{(a,d)} < P_{(b,d)}$. During the contact, in addition to the exchange of messages, the Delivery Predictability for each node can be updated.

The Delivery Predictability calculation is divided into three parts. When two nodes meet each other, they immediately update the Delivery Predictability as shown in Equation (1):

$$P_{(a,b)} = P_{(a,b)old} + (1 - P_{(a,b)old}) \times P_{init}, \quad (1)$$

where P_{init} is an initialization constant.

If, for a period of time, a pair of nodes does not encounter each other, then the Delivery Predictability metric is updated by the nodes as shown in Equation (2):

$$P_{(a,b)} = P_{(a,b)old} \times \gamma^k, \quad (2)$$

where γ is an aging constant, and k is the number of time slots elapsed since the metric was updated for the last time.

The transitive property is applied if node a frequently encounters node b , and node b frequently encounters node c . Therefore, node c presents itself as a good choice to forward messages to node a . The transitive property affects the Delivery Predictability as shown in Equation (3):

$$P_{(a,c)} = P_{(a,c)old} + (1 - P_{(a,c)old}) \times P_{(a,b)} \times P_{(b,c)} \times \beta, \quad (3)$$

where β is a constant that determines the impact of transitivity on the Delivery Predictability.

4. NECTAR

The NECTAR protocol uses the occurrence of an opportunistic contact to calculate a Neighborhood Index and spread messages in a controlled manner. During the contact period, nodes first start the transmission of messages whose destination is the node that established the contact, then exchange information about the neighborhood (Neighborhood Index), and eventually forward other messages. The spread of the Neighborhood Index allows a more accurate knowledge of network topology, which, inherently, aids the routing task. Although, to avoid unnecessary consumption of network resources, each node maintains a cache of neighbors it has contacted recently. Hence, nodes i and j can establish a new contact only after T_{slot} time slots (*units of time*) elapsed from the last contact between them, and if the Neighborhood Index table or message storage area (buffer) has changed.

The NECTAR protocol uses a movement-based heuristic. This heuristic considers that nodes in real stochastic scenarios move in a way that there is some likelihood that these nodes can meet again their past neighbors. Consequently, sending messages in a controlled way to a neighborhood of a destination increases the message delivery probability and reduces traffic on the network. The functions performed by the NECTAR protocol can be summarized in three procedures: Neighborhood Index calculation (section 4.1), Message Scheduling Algorithm (section 4.2), and Message Discard Policy (section 4.3). Table 1 defines all the NECTAR parameters, including those used during the Neighborhood Index calculation.

4.1 Neighborhood Index Calculation

The Neighborhood Index is based on recent contacts' history, in such a way that nodes that are frequent neighbors present a high Neighborhood Index. When the first contact between nodes i and j occurs, the Neighborhood Index to each other is assigned to 1, and while nodes i and j are within radio range, the Neighborhood Index and the *Contact* counter are increased in a linear fashion. Then nodes i and j update the Neighborhood Index for destinations that are not within radio range. Suppose that node j has a better Neighborhood Index to node d than node i . In this case, the node's i Neighborhood Index to node d ($N_{(i,d)}$) will be computed by the following procedure. We divide *Contact*_(j,d) counter, which represents the number of time slots that nodes j and d remain in contact by two

Table 1: NECTAR Parameters

Item	Description
$Contact_{(i,j)}$	Define the amount of time slots that i and j remained in contact
$Hops_{(i,j)}$	Express the number of hops required for i to reach j
TS	Current Time Stamp
$ts_update_{(i,j)}$	Time Stamp of the last route update from i to j
α	Define the maximum value of TTL
σ	Define the aging constant
ω	Weight applied to a known Neighborhood Index
γ	Define the maximum buffer occupancy for receiving messages during the Epidemic Level phase
$MinEpidemicLevel$	During this phase, NECTAR acts as the Epidemic Protocol
$MaxEpidemicLevel$	During this phase, NECTAR depends on the value of the parameter γ to act or not as the Epidemic Protocol
$N_{(i,j)}$	Neighborhood Index from i to destination j

metrics: a distance metric and an aging metric. The distance metric is calculated by adding 1 to $Hops_{(j,d)}$ counter, which represents the amount of hops between j and d . The amount of time slots that nodes j and d are out of radio range raised by an aging constant (σ) defines the aging metric. The Neighborhood Index formula, shown in Equation (4), favors the delivery of messages to neighbors that are near from a destination and have been in contact recently.

$$N_{(i',d)} = \frac{Contact_{(j,d)}}{(Hops_{(j,d)} + 1) \times (TS - ts_update + 1)\sigma} \quad (4)$$

In addition, if node i has already a route to node d , and node j has a better Neighborhood Index to node d , $N_{(i,d)}$ will be updated in a weighted fashion. With this approach, the Neighborhood Index calculation mitigates the impact of new information, and prevents nodes from dramatically altering a known Neighborhood Index with data that may have a limited validity. The Neighborhood Index is changed, however the associated value is reduced, allowing another neighbor, with a better Neighborhood Index, to be the next hop, as shown in Equation (5):

$$N_{(i,d)} = \begin{cases} N_{(i',d)}, & \text{if node } d \text{ is unknown} \\ \frac{(N_{(i,d)} \times \omega) + N_{(i',d)}}{\omega + 1}, & \text{otherwise.} \end{cases} \quad (5)$$

4.2 Message Scheduling Algorithm

Among other objectives of this proposal, we evaluate the behavior of routing protocols for DTNs in real stochastic scenarios. Hence, we must consider that network bandwidth is limited. It is therefore desirable that nodes select messages to be transmitted, taking advantage of the available time of a contact, and avoiding the loss of a transmission opportunity.

The Message Scheduling Algorithm procedure determines the delivery priority of each message in the storage area (buffer). Messages whose destination is the current contact are sent first. That is, if nodes i and j established a contact, node i first send messages whose destination is node j , and vice-versa. To avoid network congestion, messages are created with the TTL field assigned to a determined value, and this value is decreased by one in each intermediated node. Messages with TTL field of one will only be delivered to the

destination node. The maximum value of the TTL field is determined by the parameter α . After this first phase, the two nodes update the Neighborhood Index as described in Section 4.1, and send messages whose $NextHop$ field points to each other. After delivering a message to its destination, nodes maintain the headers' message for $tll_deleted$ time slot in a different storage area to avoid receiving redundant copies of the same message. We use this mechanism as a *PassiveAck*. Suppose that node j delivered message X to destination d and then moved toward node i , that has the same message X to node d and is out of node d 's radio range. When nodes i and j update the Neighborhood Index, node j will inform that message X to node d has already been delivered. With this information, node i will save a copy of message X 's header for $tll_deleted$ time slot and clean up the storage area.

The NECTAR protocol can operate as the Epidemic Protocol. $MinEpidemicLevel$ and $MaxEpidemicLevel$ parameters were defined to allow this mode of operation and determine how many times a message can flood the network. If the expression $(\alpha - TTL) < MinEpidemicLevel$ is true, then the message will be transmitted to all neighbors. If $MinEpidemicLevel$ parameter is set to one, only the source node will flood the network. If $MinEpidemicLevel$ parameter is set to two, the source node will flood the network, as well as the source's direct neighbors. To avoid buffer overflow problem and network congestion, when the expression $MinEpidemicLevel < (\alpha - TTL) < MaxEpidemicLevel$ is true, neighbors only accept this kind of message if the buffer occupation is below a certain threshold (γ). This configuration can also be used as a priority mechanism, i.e., nodes with specific requirements may be configured with high values of $MinEpidemicLevel$ and/or $MaxEpidemicLevel$ parameters, increasing the delivery likelihood and decreasing the delay. Nodes with different resources may be configured with different values of the parameter γ .

4.3 Message Discard Policy

Another characteristic of real stochastic scenarios considered in this paper is that nodes storage area is also limited. So, a node may not be able to deliver all messages it would like due to a neighbors storage area overflow. Hence, a Message Discard policy is essential.

Received messages are stored and associated with two parameters: M_{repl} , which counts the number of messages' replicas, and $M_{slot}(TimeStamp - ts_update)$, which represents the number of time slot elapsed since the receipt of the message. This information is used to perform the message aging index calculation.

Message discard occurs when the storage area reaches the maximum occupancy. The Message Discard Policy is intended to maintain recent messages and those that have been forwarded to few neighbors. Thus, messages with higher aging index (M_{age}) are discarded in accordance with the following formula:

$$M_{age} = M_{repl} \times M_{slot}.$$

4.4 Packet Format

In the previous sections we have shown how NECTAR protocol works. A feasible way to spread the required information to perform all described tasks is to encapsulate bundle messages into a new packet. Table 2 describes the

Table 2: Packet Format

Item	Description
<i>Source EID</i>	Endpoint Identifier of the source node
<i>Destination EID</i>	Endpoint Identifier of the destination node
<i>Sequence Number</i>	Sequence number between a source EID and a destination EID
<i>TTL</i>	This field is decremented by the source and each intermediated node
<i>MinEpidemic</i>	This field can be assigned to 0 or 1. If its value is assigned to 1, intermediate nodes should receive the message, otherwise, depending on the value of the parameter <i>MaxEpidemicLevel</i> , intermediate nodes may refuse to receive the message
<i>MaxEpidemic</i>	This field can be assigned to 0 or 1. If its value is assigned to 1, intermediate nodes should receive the message only if the buffer occupation is below a certain threshold (γ), otherwise intermediate nodes may refuse to receive the message
<i>Payload Size</i>	Determines the number of bytes of the bundle message
<i>Payload</i>	The bundle message
<i>CRC</i>	The Cyclic Redundancy Check

packet format.

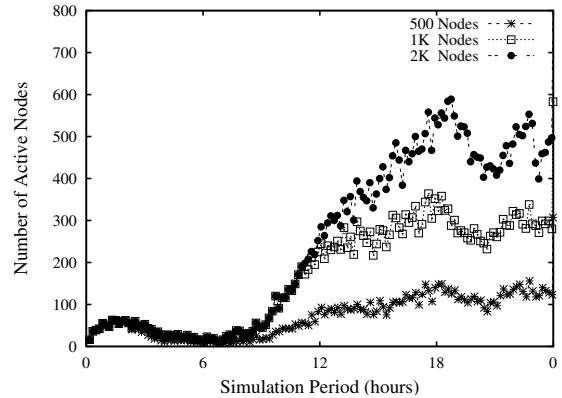
5. SIMULATION RESULTS

In our concept, the confidence on a probabilistic routing protocol evaluation depends on how realistic the mobility and traffic pattern are. The Random Waypoint Mobility Model and the Random Walk Mobility Model described in [4] are the two most common mobility models used by researchers. In the first model, a simple mobility model based on random directions and speeds is proposed, whereas in the second model, nodes randomly choose a destination, speed and a pause time. These models are most likely to be unrealistic, since real nodes in the network do not move on a completely random fashion. Instead they almost always move toward a specific destination, then toward another destination, and so on. Hence, it is desirable to use real mobility patterns for the routing protocol evaluation.

We developed a DTN network simulator to support the data format trace of real movements extracted from CRAW-DAD [7]. We also implemented the Epidemic routing protocol based on [17], the PROPHET routing protocol based on [10, 11], as well the NECTAR routing protocol. The trace-driven simulator supports only the network layer, since our objective is to evaluate routing protocols. Details about the underlying layers like error correction, error detection, collisions, and retransmissions are not implemented, so messages may be discarded only due to buffer overflow. Although this assumption is not realistic, it is applied to all protocols, which ensures a fair comparison.

5.1 Simulation Scenario

A 24-hour fragment of a real movement trace with almost 2,000 nodes and more than 47,000 records was extracted from CRAW-DAD [7]. To deal with this movement file, five different traffic data were generated with 2,000 messages distributed randomly within the first 8 hours, and a warm-up period of 600 seconds. These traffic files were used to compare the evaluated protocols. As we considered using realistic scenarios, some messages destination may be unavailable at the moment of the generation and may remain unavailable during the entire simulation. The results obtained from simulation with greater constrained resources are particularly


Figure 1: Active Nodes Average

interesting because they are closer to reality.

We want to check the behavior of the three routing protocols in different environments: nodes with storage area limited to 50, 100, 200, and 500 messages (which represents respectively 2.5%, 5%, 10% and 25% of the evaluated scenario capacity) and with the movement data limited to 500 nodes, 1,000 nodes, and almost 2,000 nodes. Figure 1 shows the average of active nodes in the interval of 600 seconds during the period of the simulation (24 hours). We only consider nodes that turn on their radio during the simulation. In the scenario with 500 nodes, the maximum and the average active nodes correspond to 61.4% and 14.5%, in the scenario with 1,000 nodes they correspond to 58.3% and 17.4%, and in the scenario with 2,000 nodes they correspond to 41% and 12.7%, respectively. We can note that the average active nodes during the simulation period is very small (nodes turn on and off several times during the simulation period), and this observation will be useful at the time we analyze message delivery rate.

The delivered messages, i.e., the number of messages that a protocol is able to deliver to the destination, is one of the most important metrics. The analysis of the number of hops required for a message to be delivered to its destination and the number of messages exchanged can reveal the knowledge that nodes have about a network. Even though DTN applications should be delay-tolerant, the delay metric is also important. The results presented in this paper are based on the average of simulations with five different traffic data, and the bars represent the 95% confidence interval.

5.2 Parameter Adjustment

In order to adjust the NECTAR Routing protocol parameters, we configured a special scenario with high resource constraints. Traffic was randomly distributed during the 24 hours of simulation, the movement data was limited to 500 nodes, and we vary the parameters *MinEpidemicLevel*, *MaxEpidemicLevel*, ω , σ , and γ . In the first set of simulations we vary the parameter ω from 0 to 6, the parameter σ from 0.05 to 0.95 in intervals of 0.05, and set *MinEpidemicLevel*, *MaxEpidemicLevel*, and γ parameters to 1. The simulation results indicates that if we set the weight parameter (ω) to 5 and the aging constant (σ) to 0.05 we maximize the number of delivered messages. With this configuration ($\omega=5$), we prevent nodes from dramatically altering a known Neighborhood Index with data that

may have a limited validity. At the moment node j leaves node i 's radio range, the Neighborhood Index for each other is reduced, and this information is immediately propagated to their neighbors. The impact of this configuration on the Neighborhood Index reflects the capacity of learning new routes quickly and taking advantage of the contacts.

The reference to the term *EpidemicLevel* means that the *MinEpidemicLevel* and *MaxEpidemicLevel* parameters have the same value. We decide to investigate the impact of the *EpidemicLevel* phase varying it from 1 to 4, with the γ parameter assigned to 1, which means that neighbors should receive all messages unconditionally. The results show that with high resource constraints (storage area limited to 50 messages) the *EpidemicLevel* = 1 obtained the best delivery rate and exchanged fewer messages. This result proves that flooding the network without any control does not produce good results, due to buffer overflow and constant message broadcast.

As we increase the node's storage area to 100, 200 and 500 messages, *EpidemicLevel* equal to 4 presents the best delivery rate. However, the number of messages exchanged grew to an undesired value (about 90% of the Epidemic Routing Protocol results). This growth occurred due to the message flooding during *EpidemicLevel* phase. Based on this conclusion, we decided to propose a routing protocol that should be able to dynamically adapt to the network and nodes limitations. Suppose a network configuration with *MinEpidemicLevel* assigned to 2, *MaxEpidemicLevel* assigned to 4, and γ assigned to 0.90. For the first two hops, nodes will perform message flooding and the neighbors will receive them. For the third and the fourth hops, neighbors may refuse messages in the case the storage area occupancy is above 90%.

To investigate the behavior of the NECTAR Protocol we run simulations setting the *EpidemicLevel* parameters using the format (*MinEpidemicLevel*, *MaxEpidemicLevel*) as: (1,2), (1,3), (1,4), (2,3), (2,4), and (3,4). The γ parameter has been associated with values varying from 1.00 to 0.50, with a step of 0.05, which implies the provision of 100%, 95%, 90%, and so on of the storage area for messages transmitted during the *MaxEpidemicLevel* phase. The configuration with *EpidemicLevel* assigned to (1,4) and γ assigned to 0.90 presents the best results. We observed that the increment of *MinEpidemicLevel* does not produce good results, and limiting *MaxEpidemicLevel* to 4 balances the relationship between delivered and exchanged messages.

5.3 Results

We present a comparative simulation analysis of the NECTAR, Epidemic and PROPHET routing protocols. Each protocol was evaluated in scenarios with medium and high resource constraints. The characteristics of the scenarios and protocols are summarized in Table 3.

The storage area will be identified as follows: 50B, 100B, 200B, and 500B will describe the storage area for 50, 100, 200, and 500 messages, respectively. The scenarios used in the simulations will also be identified as follows:

1. scenario 1 describes the environment with 500 nodes;
2. scenario 2 describes the environment with 1,000 nodes;
3. scenario 3 describes the environment with almost 2,000 nodes.

Figure 2 shows the delivered messages for all protocols

Table 3: Scenarios and Protocols Configuration

Scenarios Configuration	
Configuration	Description
<i>Active Nodes</i>	500, 1,000, and almost 2,000 nodes
<i>Storage Area capacity</i>	50, 100, 200, and 500 messages
<i>Traffic Distribution</i>	during the first 8 hours
<i>Simulation Period</i>	24 hours
<i>Simulator TIC</i>	50 ms
Epidemic Routing Protocol	
<i>Contact Update</i>	500 ms
<i>Forwarding Policy</i>	Flooding
<i>Discard Policy</i>	FIFO
PROPHET Routing Protocol	
<i>Contact Update</i>	500 ms
<i>Forwarding Policy</i>	GRTR
<i>Discard Policy</i>	MOFO
<i>P.Encounter</i>	0.75
β	0.25
γ	0.99
NECTAR Routing Protocol	
<i>Contact Update (T_{slot})</i>	500 ms
ω	5
σ	0.05
γ	0.90
<i>MinEpidemicLevel</i>	1
<i>MaxEpidemicLevel</i>	4

with scenarios 1 and 3. The results show that PROPHET protocol delivered more messages than Epidemic protocol with high level of resource constraints (50B), but as storage area capacity grows, the delivery rate of Epidemic exceeds PROPHET in more than 155% in scenario 1, and more than 80% in scenario 3. The PROPHET protocol was unable to deal with the dynamism and realism of movement and traffic data, even though we have implemented the least restrictive forwarding policy. This problem explains the low amount of delivered messages. The Epidemic protocol has behaved as expected, that is, as restrictions decrease, the number of delivered messages increase. The NECTAR protocol delivered more messages than the others protocols, regardless of storage area capacity. The Neighborhood Index formula adapts quickly to the network topology changes, providing a better adjustment to the dynamism of a real network, and achieving the goal of delivering more messages than the other evaluated protocols. Table 4 presents the number of delivered messages for all scenarios, where ER stands for Epidemic Routing, PR stands for PROPHET Routing, and NR stands for NECTAR Routing Protocol.

As explained in Section 5.1, the real movement traces used in simulation presents real characteristics, such as: users turn the mobile nodes on and off, users send messages to others users that are disconnected, and users move to areas without coverage and remain there. These real characteristics contribute to the low delivery rate with limited storage area. Although the objective of this work is to evaluate scenarios with resource constraints, results extracted from simulations with scenario 3 (least restrictive) and with infinite storage area indicates that Epidemic delivered $\approx 85\%$, PROPHET delivered $\approx 33\%$, and NECTAR delivered $\approx 75\%$ of the 2,000 messages.

Looking at the exchanged messages table (Table 5), it is clear that Epidemic protocol sends much more messages than the others proposals, as expected. This behavior explains the low amount of delivered messages in scenarios with high resource constraints due to network congestion and buffer overflow. A second important observation is that, although the NECTAR protocol has delivered more

messages that the PROPHET protocol, the amount of messages exchanged by both protocols remained at the same level. The same analysis can be applied to the discarded messages table (Table 6).

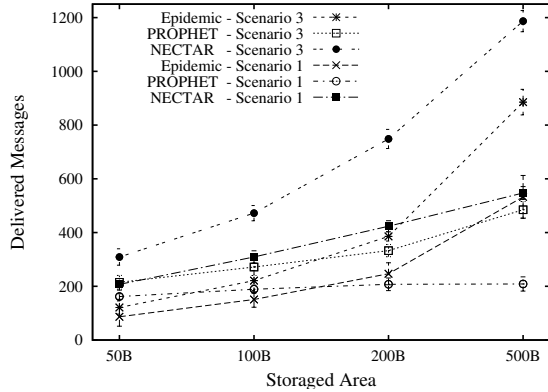


Figure 2: Delivered Messages - Scenarios 1 and 3

Table 4: Summary of delivered messages

Storage Area	Delivered Messages								
	Scenario 1			Scenario 2			Scenario 3		
	ER	PR	NR	ER	PR	NR	ER	PR	NR
50	86	162	207	112	216	295	121	214	309
100	150	189	309	199	283	461	221	271	472
200	246	207	423	345	345	710	385	332	748
500	532	208	547	778	439	1,116	885	484	1,186

Table 5: Summary of exchanged messages

Storage Area	Exchanged Messages $\times 10^6$								
	Scenario 1			Scenario 2			Scenario 3		
	ER	PR	NR	ER	PR	NR	ER	PR	NR
50	0.7	0.06	0.04	2.1	0.3	0.1	3.2	0.4	0.2
100	1.5	0.06	0.07	4.9	0.4	0.2	7.2	0.8	0.4
200	2.9	0.03	0.09	10.6	0.6	0.3	16.1	1.2	0.6
500	4.3	0.02	0.12	19.5	0.3	0.4	32.3	1.1	0.8

Table 6: Summary of discarded messages

Storage Area	Discarded Messages $\times 10^6$								
	Scenario 1			Scenario 2			Scenario 3		
	ER	PR	NR	ER	PR	NR	ER	PR	NR
50	0.67	0.05	0.02	2.1	0.2	0.1	3.1	0.4	0.2
100	1.45	0.04	0.03	4.8	0.3	0.1	7.0	0.7	0.3
200	2.80	0.01	0.03	10.4	0.5	0.2	15.8	1.0	0.3
500	4.17	0.00	0.11	19.0	0.1	0.1	31.6	0.6	0.3

Figures 3 and 4 present a CDF (Cumulative Distribution Function) of the message delivery delay and hops based on the first of the five traffic files. Due to the nodes' movement pattern and the constrained resources, Epidemic and NECTAR routing protocols increase substantially the amount of delivered messages only after the first 7 hours, when the number of active nodes start to grow. In scenario 3, NECTAR takes 17h24min and 14h of the simulation period (24 hours) to deliver the same amount of messages that Epidemic and PROPHET delivered after 24 hours, respectively. In scenario 1, NECTAR takes 12h42min and 17h42min to deliver the same amount of messages that Epidemic and PROPHET delivered after 24 hours, respectively.

The PROPHET routing protocol takes, approximately, 12 hours to increase substantially the amount of delivered messages. The Delivery Predictability calculation is more conservative than Neighborhood Index formula, so PROPHET loses several contacts opportunities which contributes to the low amount of delivered messages. Looking at Figure 4, in scenario 3 Epidemic requires 45 hops to deliver 90% (792 messages) of its messages and PROPHET requires 13 hops (413 messages). To deliver the same amount of messages, NECTAR requires only 7 and 6 hops, respectively. In scenario 1, Epidemic requires 18 hops to deliver 90% (75 messages) of its messages and PROPHET requires 10 hops (140 messages). To deliver the same amount of messages, NECTAR requires only 3 and 6 hops, respectively.

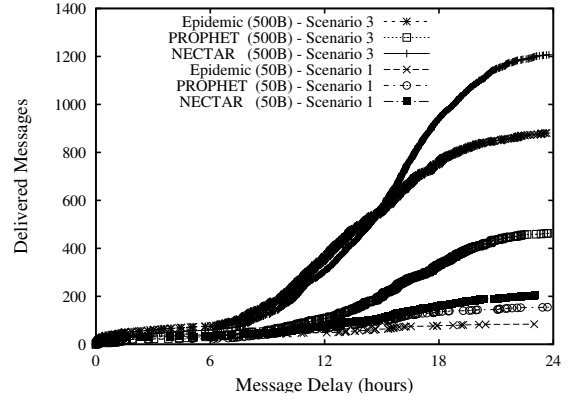


Figure 3: Message Delay CDF

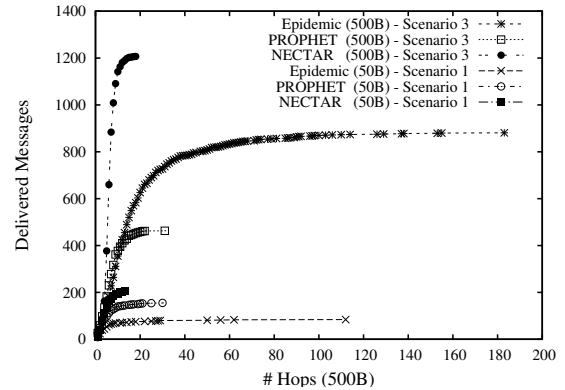


Figure 4: CDF Hops

6. RELATED WORK

In the last few years we have observed an increased interest in DTN research that leads to new theoretical models and routing algorithms. In order to establish a relationship between performance and the knowledge that nodes have about a DTN, Jain et al. [6] proposed a set of elements, called oracles, with the ability to provide DTN information for the routing algorithms. The Epidemic Routing and its variations are studied in [18], and the authors developed a framework based on Ordinary Differential Equations (ODE). In [9], the authors proposed a new routing scheme (PR-CD)

based on PROPHET protocol and studied the performance of routing schemes. The Spray and Wait [15] routing protocol uses an opportunist approach to estimate the number of nodes and get the optimal number of messages copies. The number of copies of a message is restricted to n , where n is calculated based on the number of nodes in the network. Zhao et al. [19] describe an alternative routing protocol for DTN, known as Message Ferrying (MF), where mobile devices are controlled, and may change the direction of movement in accordance with the needs of data sources.

The MaxProp [2] and RAPID [1] routing protocols were deployed on a vehicular DTN testbed called UMassDieselNet. This network consists of buses carrying 802.11b radios and a computer that intermittently establish a contact with each other, and covers a 150 square-mile area around Amherst, M.A. MaxProp classifies messages based on a cost (*delivery likelihood*) assigned to each destination, and uses acknowledgments to notify message deliveries. Rapid can optimize a specific routing metric by treating DTN routing as a resource allocation problem. A per-packet utility determines how packets should be replicated.

7. CONCLUSION

In this paper we present NECTAR, a DTN Routing Protocol Based on Neighborhood Contact History. The objective of our proposal is to increase the message delivery rate using traces of real scenarios with high constrained resource nodes. Unlike others previous works, our evaluations are based on traces of real movements extracted from CRAWDAD [7]. Simulations demonstrated that NECTAR was able to fulfill its objective of deliver more messages, exceeding Epidemic and PROPHET routing protocols in 140% and 27% in scenario 1 (most restrictive), in 43% and 154% in scenario 2, and in 34% and 145% in scenario 3 (less restrictive), respectively. NECTAR also transmits much less messages than Epidemic, and almost the same amount as PROPHET. This implies that the performance difference between NECTAR and Epidemic would be much greater if we consider the characteristics of the link layer. This proposal intends to fill the lack of a DTN protocol that can increase the number of delivered messages, consuming few network resources, in high resource constrained networks.

We plan to expand our work in a couple of directions. It is clear that nodes with high limited resources suffer continually from storage area overflow. This problem implies discarding a large amount of messages and, eventually, the network becomes unable to deliver messages. To mitigate this problem, we intend to investigate other mechanisms to prevent future transmissions of delivered messages and for releasing space in the storage area. We are also encouraged by the results obtained from our simulations to study the behavior of the NECTAR protocol using mobility traces from Vehicular DTN [16], and to compare with others proposals like [1, 2].

8. ACKNOWLEDGMENTS

This work was partially supported by IBGE, Capes, CNPq, FAPERJ, and TBE.

9. REFERENCES

- [1] A. Balasubramanian, B. N. Levine, and A. Venkataramani. DTN Routing as a Resource Allocation Problem. In *Proc. ACM Sigcomm*, 2007.
- [2] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *Proc. IEEE Infocom*, pages 1–11, 2006.
- [3] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss. Delay-tolerant networking: an approach to interplanetary internet. *IEEE Communications Magazine*, 41(6):128–136, 2003.
- [4] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. In *Proc. IEEE IWCMC*, 2(5):483–502, 2002.
- [5] A. Doria, M. Uden, and D. P. Pandey. Providing connectivity to the saami nomadic community. In *Proc. Open Collaborative Design for Sustainable Development*, 2002.
- [6] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. In *Proc. ACM Sigcomm*, pages 145–158, 2004.
- [7] D. Kotz, T. Henderson, and I. Abyzov. Downloaded from http://crawdad.cs.dartmouth.edu/dartmouth/campus/syslog/01_04, Dec. 2004.
- [8] J. Leguay, T. Friedman, and V. Conan. Evaluating mobility pattern space routing for DTNs. In *Proc. IEEE Infocom*, pages 1–10, 2006.
- [9] C.-S. Lin, W.-S. C. and; Ling-Jyh Chen, and C.-F. Chou. Performance study of routing schemes in delay tolerant networks. In *Proc. IEEE AINAW*, pages 1702–1707, 2008.
- [10] A. Lindgren and A. Doria. Probabilistic routing protocol for intermittently connected networks. RFC Draft, Internet Engineering Task Force, Feb 2008.
- [11] A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. In *Proc. ACM SigMobile*, 7(3):19–20, 2003.
- [12] OLPC. One laptop per child, 2008. Available at: <http://laptop.org/>.
- [13] R. Ramanathan, R. Hansen, P. Basu, R. Rosales-Hain, and R. Krishnan. Prioritized epidemic routing for opportunistic networks. In *Proc. ACM MobiOpp*, pages 62–66, 2007.
- [14] L. Song, D. Kotz, R. Jain, and X. He. Evaluating location predictors with extensive wi-fi mobility data. *Proc. ACM SigMobile*, 7(4):64–65, 2003.
- [15] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proc. ACM Sigcomm WDTN*, pages 252–259, 2005.
- [16] UMass. Umass trace repository, 2007. Available at: <http://traces.cs.umass.edu/>.
- [17] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-200006, Duke University, April 2000.
- [18] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. Performance modeling of epidemic routing. *Computer Networks*, 51(10):2867–2891, 2007.
- [19] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *Proc. ACM MobiHoc*, pages 187–198, 2004.