Routing Protocols for Delay Tolerant Networks: a Quantitative Evaluation

Khalil Massri
Dipartimento di Informatica e Sistemistica
Università degli Studi di Roma
Via Ariosto 25 - 00185 Roma, Italy
massri@dis.uniroma1.it

Alessandro Vernata
Dipartimento di Informatica e Sistemistica
Università degli Studi di Roma
Via Ariosto 25 - 00185 Roma, Italy
alex.vernata@gmail.com

Andrea Vitaletti
Dipartimento di Informatica e Sistemistica
Università degli Studi di Roma
Via Ariosto 25 - 00185 Roma, Italy
vitale@dis.uniroma1.it

ABSTRACT

In this paper we propose a new taxonomy for Delay Tolerant Networks (DTNs) routing protocols and a thorough quantitative evaluation of many protocols proposed in the literature.

We categorize DTN protocols, according to their use of three main techniques: queue management, forwarding and replication. Queue management orders and manages the messages in the node's buffer, forwarding selects the messages to be delivered when there is a contact and finally replication bounds the number of replicas in the network.

Contrary to most previous papers, where either only qualitative comparisons have been presented or only a single category of protocols have been analysed, in our work we discuss the results of our experimental activity on many of the DTN protocols in the literature. Our results show that, an effective combination of the proposed techniques, can significantly improve the performance of the protocols in terms of delivery ratio, overhead and delay.

Categories and Subject Descriptors

H.4 [**Network Protocols**]: Routing protocols; D.2.8 [**Store and forward networks**]: [evaluation, performance measures]

General Terms

Experimentation, Performance

Keywords

Delay Tolerant Network

1. INTRODUCTION

Delay tolerant networks [13] (DTNs) are made of wireless nodes characterised by high end-to-end latency, frequent disconnections, limited resources (e.g., battery, computational power, bandwidth), and unreliable wireless transmission. However, nodes in a DTN are able to communicate even if a route connecting them never exists. To this purpose, beyond wireless transmission, DTNs also use nodes' mobility in order to carry the message to the intended destination. DTNs are often disconnected, and thus nodes are not supposed to possess or acquire any knowledge about the network topology, which is instead necessary in traditional MANET routing protocols. As possible examples of contexts in which DTNs might operate we mention: Inter-planetary networks [8], DFT-MSN for pervasive information gathering [36], Social DTN [16][9] and Ferry Boats [38].

Due to the wide diversity in contexts in which DTN routing is applicable, many different protocols have been proposed in the literature.

Contribution of the paper The first contribution of this paper, is the clear identification of the main techniques characterising DTN routing protocols in order to better understand and classify the solutions proposed in the literature. In particular, we consider *forwarding*, *replication* and *queue management* as the main techniques for DTN protocols and we show how each protocol can be classified according to the techniques it adopts. We observe that those technique has been already considered in the literature, but we propose a unifying approach which in our opinion better clarify their role and interaction (see Figure 1)

The second contribution is the first quantitative comparison of many DTN routing protocols proposed in the literature based on extensive simulations with the ONE simulator [18]. To achieve this goal, we extended the ONE simulator, which is considered one of the reference simulator for DTN [23], implementing new mobility models and DTN protocols. We stress again, that to the best of our knowledge, this is the first paper that tries to provide a quantitative comparison.

Structure of the paper The paper is organised as follows: in section 2 we briefly present previous works surveying DTN routing protocols and we clarify our contribution with respect to the state of the art. In our approach, we abstract the most relevant techniques adopted by DTN protocols in the literature, namely forwarding, queue management and replication and in section 3 we use such techniques to define a taxonomy for the classification of DTN routing protocols. Finally, in section 4 we present the results of our experimental activity.

2. RELATED WORK

MSWiM'12, October 21–25, 2012, Paphos, Cyprus.

Various surveys have been proposed in order to agree a common taxonomy on DTN routing protocols, and to provide, in some cases, an evaluation framework [26] or guidelines [33] useful to identify the main features best supporting specific user requirements.

The simplest taxonomy proposed in [28] categorizes the protocols in ad-hoc and infrastructure ones. Ad-hoc protocols (do not rely on any infrastructure), are further classified into context-based and dissemination-based.

Most of the surveys [33, 4, 24, 5, 31], consider the knowledge about the network as a distinguish feature to classify DTN protocols and to decide whether *forwarding* a message or not. Under this perspective, protocols based on flooding are the most basic ones, because they do not assume any knowledge; they always forward available messages. More advanced protocols, forward the messages on the basis of information such as, the location [25], link metrics [17] either acquired interacting with an infrastructure [15] or through the exchange of information with occasional encounters [39][9]. In [2] and [22] the knowledge about the network is modelled either through deterministic process or stochastic ones. Some recent papers, also consider the underlying social network of users to improve the effectiveness and efficiency of DTN protocols. [23] [26] analyse a variety of social protocols, and proposes a classification technique that also consider social information as a key element to characterize DTN protocols.

Some papers in the literature consider *replication* to improve the effectiveness of the proposed protocols [21] and classify the protocols according to their replication policies [26]. In [20], *queue management* is considered a fundamental technique to improve DTN protocols' performance and [19] discusses a queue management architecture for DTNs.

In our paper, we deliberately do not consider infrastructure DTN protocols, rather we focus on ad-hoc DTN routing protocols. In our approach, we first abstract the most relevant techniques adopted by the DTN protocols in the literature. We then show that each DTN protocol is either based on a single technique, or on the combination of some techniques. As a consequence, each protocol can be classified in our taxonomy according to the techniques it adopts. In particular, we started considering the techniques proposed in the taxonomy presented in [33], and we identified *forwarding* and *replication* as the most distinguish ones. Furthermore, in light of the work of Lenas et al. [19], we also consider *queue management* as a fundamental technique.

To the best of our knowledge, this is the first paper that tries to provide a quantitative comparison of many DTN routing protocols proposed in the literature. We remark that previous surveys, only provide qualitative comparison of the protocols; as an example, see Table 2 in [33] and Tables 1, 2 and 3 in [5]. On the contrary, our results are based on extensive simulations with the ONE simulator [18] which is considered one of the reference simulator for DTN [23].

3. THE PROPOSED TAXONOMY

In this section we first discuss the three techniques used in our taxonomy to classify the DTN routing protocols, we then classify the main protocols proposed in the literature accordingly.

3.1 Techniques

Figure 1: Node Architecture

The simplified internal architecture of a DTN node is shown in Figure 1. Upon a contact, a node exchange a summary of the information necessary to update its knowledge (i.e. a local view) on the environment in which it operates. As an example, it can exchange the list of messages in the queue or the list of prior encountered nodes. This knowledge drives all the subsequent actions and is used by the queue management (QM) to assign a rank to the messages in the queue. If the queue is full, queue management applies a suitable eviction policy to make room for new incoming messages. The forwarding policy (FW) selects the messages to be forwarded which are possibly duplicated (R) in order to improve the effectiveness of the protocol.

Queue Management.

Queue Management (QM) defines a total ordering on the messages in the queue on the basis of the node's knowledge. QM orders all the messages, also those ones that will not be taken into consideration as possible candidate to be forwarded. When new messages must be accommodated in the queue and there is no room, the eviction policy selects the messages to be discarded according to that ordering. The intrinsic delay of DTN applications implies that messages can be enqueued for a long time. Due to the limited buffer size of DTN nodes, an efficient queueing management policy must be adopted to avoid the eviction of important messages. In the following, we briefly discuss the main QM policies.

- First-In-First-Out. FIFO is the default QM policy.
- Destination Independent. Where, QM uses only parameters that are not related to the destination, such as message hop count [1], forward transmission count and message size [14], and other types listed in [20].
- Destination Dependent. In this case, QM uses parameters dependent on the destination. In [1] the cost to reach the destination is computed using Dijkstra while [3] estimates the delay. [36] uses the message fault tolerance which is an indicator of the importance of the fact that a message reaches its destination (this is related to the redundancy of the message).

Forwarding.

Forwarding (FW) selects the subset of messages in the queue to be forwarded. In most cases, the messages are selected on the basis of the same ordering defined by the QM, but in some cases [1] [3] a new ordering among messages can be employed. Contrary to what happens for QM, this new ordering is defined only with respect to the current contact and last only for the contact time (i.e. the time the two nodes can communicate). In other words, the new ordering is only used to define a priority on the messages to be forwarded to the current contact.

- Direct Delivery. The message is only delivered to the destination (i.e. no intermediate nodes are employed)
- Always. Upon a contact, the message is always forwarded. This policy, requires low computation power, but can lead to a very high number of transmissions.

- Knowledge based. Messages are forwarded based on the current knowledge in terms of contextual, historical or social information. Contextual information regards the knowledge that can be inferred by the node only on based on its current status such as battery level, speed, direction etc. Historical information are obtained over time and are used to evaluate future behaviours of the network; inter-contact time, history of encounters, contact time, are examples of such information. Finally, considering that in many DTN scenarios nodes are carried by humans, social information are used to describe the relationships among the users in order predict social behaviours that can be used to improve the effectiveness of the forwarding policy.

Replication.

Replication (R) controls and bounds the number of copies of a message in the network and is used to increase the robustness of the protocols. Observe that a message that has been selected by the FW policy to be delivered to the current contact is normally deleted from the queue, but the R policy, can enqueue it again in order to have multiple copies into the network.

1. Single copy. Messages are never replicated. Once a message has been delivered to an encounter it is deleted from the queue.
2. Limited. The total number of copies of a message in the network is bounded.
3. Controlled. A message is replicated only if a condition holds.
4. Unlimited. There are no constrains on the number of replicas in the network.

3.2 Classification of DTN Routing Protocols

In this section we classify most of the DTN routing protocols presented in the literature according to their use of the proposed techniques. The result of this effort is shown in Table 1.

Direct Delivery and First Contact[18]. They are single copy routing protocols, namely only one copy of each message exists in the network. In Direct Delivery, messages are only delivered to their final destination. First Contact always forwards the messages to the first contact and then deletes them from the queue; this means that messages can be handled by several nodes. Both protocols employ FIFO queueing.

Epidemic Protocol[35]. Messages are broadcasted to all neighbours. When there is no room in the message queue, oldest message are evicted. Messages are always forwarded according to a FIFO policy and no bound on the number of replicas are considered (i.e. R is unlimited).

Spray and Wait Protocol[32]. This protocol is made of two phases. Spray phase (only once): L message copies are initially spread to L distinct "relays". Wait phase: if the destination is not reached in the spray phase, the L nodes carrying a message copy perform direct transmission. The forwarding technique is a mix of always (Spray) and direct delivery (Wait). The replication is limited (the bound is L) and the queue management is FIFO.

[1] D.D: Direct Deliver Protocol; F.C: First Contact Protocol

PROPHET[21]. PROPHET uses delivery predictability as a metric to evaluate the likelihood of one node to reach the destination. A node carrying a message deliver it to all the neighbours with delivery predictability higher than its one. The queueing and replication techniques are the same as Epidemic (FIFO and unlimited), but forwarding is based on delivery predictability and it is thus history based. Similar considerations on forwarding hold for protocols like FRESH (that use last encounter age) [12] and SEPR [34] (that evaluate the shortest path).

Fuzzy Spray[14]. The protocol is based on two parameters: the Forward Transmission Count (FTC) and the message size. These parameters are input to a fuzzy rule which prioritizes the messages to be transmitted. Selected messages are broadcasted to all the neighbours. The forwarding and replication techniques are the same as Epidemic (always and unlimited). The queue management is destination independent and is based on the message priority.

SCAR[27]. Context and history information, such as colocation, mobility and battery level are used to calculate the delivery predictability of each neighbour and messages are delivered to neighbours with highest delivery predictability first. For each message, there is a single master and L backup copies generated by the source node. Messages are ordered in the queue according to their master/backup label and master messages are forwarded first. Backup copies can be evicted, while the master is never evicted. In this case, forwarding is a mix of context and history based, replication is limited and queue management is destination independent.

FAD[36]. Similarly to PROPHET, FAD forwards the messages to the neighbours with highest delivery probability. However, in this case messages are ordered in the queue according to their fault tolerance. Fault tolerance is proportional to the number of replicas in the network, and also take into consideration the delivery probability. Messages with low fault tolerance are forwarded first in order to increase the number of copies in the network of messages with low probability of reaching the destination. In this case, forwarding is history based, while queue management is destination dependent and replication is unlimited.

MaxProp[1]. Each node has a routing table which predicts the likelihood (i.e. the cost) to reach another node in the future through its current neighbours. Routing tables are updated on the basis of the information obtained by the neighbours and messages are ordered and forwarded according to their cost to reach the destination. As such, forwarding is history based and replication is unlimited. The queuing technique is a combination of destination dependent (Dijkstra to evaluate the cost to reach the destination) and destination independent (the current number of hops that a message has accumulated).

RAPID[3]. Most DTN protocols aim at maximizing the probability of finding a path between the source and the destination, neglecting all the other relevant metrics such as delay, energy consumption etc. On the contrary, RAPID tries to optimise a specific routing metric (e.g., worst-case delivery delay). RAPID treats DTN routing as a resource allocation problem. It translates routing metrics into per-packet utilities which determine how packets should be replicated in the system. When there is need to make room for new messages, messages with lower utility are deleted first. Forwarding technique is history based, messages with higher marginal utility are forwarded first and replication

		Protocols															
		D.D[1]	F.C	[35]	[21]	[32]	[14]	[27]	[36]	[1]	[3]	[10]	[11]	[30]	[7]	[9]	[16]
FW.	Direct Del.	√	-	-	-	√	-	-	-	-	-	-	-	-	-	-	-
	Always	-	√	√	-	√	√	-	-	-	-	-	-	-	-	-	-
	Context	-	-	-	-	-	-	√	-	-	-	-	-	√	√	-	-
	History	-	-	-	√	-	-	√	√	√	√	√	√	-	√	-	-
	Social	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	√
R.	Single	√	√	-	-	-	-	-	-	-	-	√	-	-	-	√	-
	Limited	-	-	-	-	√	-	√	-	-	-	-	-	-	-	-	-
	Controlled	-	-	-	-	-	-	-	-	-	-	-	√	√	√	-	√
	Unlimited	-	-	√	√	-	√	-	√	√	√	-	-	-	-	-	-
Q.M.	FIFO	√	√	√	√	√	-	-	-	-	-	√	-	-	√	√	√
	Dest. Dep.	-	-	-	-	-	-	-	√	√	√	-	-	-	-	-	-
	Dest. Indep.	-	-	-	-	-	√	√	-	√	-	-	√	√	-	-	-

Table 1: Protocols Classification

is unlimited. Queue management is destination dependent, messages are prioritized according to their utility value (i.e. the estimated delay to reach the destination).

Cluster-Based Routing[10] The basic idea of this protocol, is to distributively group mobile nodes with similar delivery probability (i.e. nodes that are frequently in contact) into a cluster. Nodes in a cluster can interchangeably share their resources (such as buffer space) for overhead reduction and load balancing. This protocol uses single copy replication, FIFO queueing, and its forwarding technique is history based (cluster-based).

NETCAR [11] The scheduler of NECTAR uses the Neighbourhood Index (contact history) to select the messages to be forwarded. Message replication is controlled using parameters that comes from the context (buffer occupancy, maxTTL, minEpidemicLevel, maxEpidemicLevel, etc.). Message ageing and number of replicas are used to evict messages. NETCAR uses an history based forwarding with controlled replication and destination independent queue management.

ORWAR [30] This protocol uses context information such as speed, direction, radio range and bandwidth to estimate the contact window size. Only messages that can be transmitted in such interval are forwarded. A message utility function (defined by the application), the so called "utility per bit ratio", ranks the messages in the queue and also regulates the number of replicas for each message. Forwarding is context based, replication can be considered controlled, and queue management is destination independent.

HiBOp [7] In HiBOp, a mix of context and history information is used to forward messages only to nodes with higher delivery probability. This knowledge, is also used to control the amount of message replicas (a measure of redundancy and fault tolerance for the message is computed). Only the source can replicate a message, the other nodes can only forward it. The number of replicas is calculated such that the probability of losing all the replicas is below the threshold specified by the application. This implies that replication is controlled and forwarding is a mix of context and history based.

BubbleRap [16] and SimBet [9] They are both social based protocols. BubbleRap combines the knowledge on the community structure and centrality of the nodes to decide whether to forward. Each message has two types of ranking, global and local (related to its community). Forwarding is done using the global ranking, until the message reaches a node which is in the same community of the destination node. Then, local ranking is used until the destination is reached or the message expires. Each node forwarding a message do not delete its copy unless the message is delivered to the community of its destination. In this case, forwarding is social based and replication is unlimited. In SimBet similarity and betweenness utility functions are combined to select the message to be forwarded and there is only a single copy of the message in the network (i.e. no replication)

4. EXPERIMENTAL RESULTS

In this section we discuss the experimental results to compare the performance of many DTN routing protocols proposed in the literature. We stress again that, to the best of our knowledge, this is one of the first attempt to quantitatively compare the performance of many DTN protocols. We used The Opportunistic Network Environment simulator (The ONE)[18] and we extended it in order to support new protocols and mobility models.

4.1 Scenario Settings and Metrics

The main purpose of this paper is to compare the performance of most of the DTN protocols proposed in the literature. To this purpose, we considered a reference scenario which is general enough to be not suitable for only specific protocols, but at the same time will be based on realistic technical assumptions. For the sake of simplicity, we assume the nodes of the network are mobile phones carried on by humans moving in an area of 400x200 meters (e.g. a square in a city). In this context, there are two main wireless technologies widley available for ad-hoc connectivity on mobile phones: WiFi and Bluetooth. In our simulation, we consider Bluetooth, because it is the only one with a reference implementation supporting the deployment of DTN protocols [**?**]. According to Bluetooth specifications, transmission speed is about 0.2 Mbps = 25.00kBps and transmission range about 10m. The average speed of nodes (used in deferent mobility models) is chosen to be in the range of walking speed (2,8 - 7,2 Km/h).

To evaluate the effects of the buffer size on the performance, we combined different message and buffer sizes. In the small buffer size case, there are an average of 70 messages in the queue, while in the larger case the messages are about 700. We vary the number of nodes from 10 to 70 nodes moving in the reference area. Each node generates a new message to a random destination every 1-3 minutes. Messages never expire, namely their TTL is infinity.

Nodes move according to four types of mobility models as disscused in section 4.2. The settings used to run our simulations are summarised in Table 2.

Simulation time	43200s (12h)
Transmission speed	25kBps
Transmission range	10m
Node speed	0.8 - 2.0 m/s
Message size	100B - 200B
Message creation interval	[60 - 180]s
Buffer size	10kB, 100kB
Number of nodes	10,20,40,70

Table 2: Simulation Settings

The metrics used to compare the performance of the protocols are described in the following:

- *Delivery Ratio* is the ratio between the delivered messages over the generated ones.
- *Overhead* is the ratio between the total number of transmissions over the number of delivered messages.
- *Average Delay* is the average time needed for a messages to reach its destination.

Those metrics have been used not only to evaluate the performance of the DTN protocols, but also to evaluate to which extent, the proposed techniques, namely QM, FW and R, impact on those performance. The results are obtained varying the above parameters for each protocols and averaging the results over five runs.

4.2 Discussion

In this section we discuss the experimental results obtained simulating most of the protocols discussed in section 3. In particular we consider three classes of protocols, simple, single technique, and advanced protocols. For each class, we discus how and why the proposed techniques affect the performance metrics.

Simple protocols.

Simple protocols do not use knowledge information in managing the messages. As a consequence, they are the less demanding protocols in terms of computational capabilities and have been among the first protocols proposed in the literature. Direct Delivery, First Contact [18] and Epidemic [35] are the protocols that we analyse in such class.

Epidemic explores all possible options, namely all nodes are possible carriers. This implies, that when buffer constraints are negligible (see 100KB in Figure 2), it is the best performer in terms of delivery ratio and delay. On the contrary, when the size of the buffer becomes critical (10KB), as the density of the network increases the delivery ratio quickly decreases due to messages dropping. As a consequence, a number of messages need to be sent again and the overhead increases. We stress that this result, makes clear the importance of an effective Queue Management policy when nodes' buffer size is small.

In the considered scenario, First Contact and Direct Delivery show the same performance irrespectively from the buffer size, which is in any case sufficient to store all the messages in the network at any given time. The only difference between the two protocols, is the overhead, which is higher in First Contact because it forwards the message to the first encountered node until the destination is reached.

We remark that the only difference between the two protocols is the forwarding policy, but this difference, does not have any impact on delivery ratio and delay. We will see in the next sections that knowledge-based forwarding policies can actually improve the performance also when limited replication is employed.

Single technique Protocols.

The main objective of this section is to evaluate how the performance of the simple protocols can be improved using just one of the proposed techniques (see Figure 3). As such, the protocols analysed in this section, implement either QM or FW or R.

Spray and Wait [32] implements the replication technique. It shows good performance except for delay. The average overhead is bounded by the number of copies created in the spray phase (L=4 in our experiments). In the wait phase, those copies have to reach the destination in a direct delivery fashion. For this reason, the delay of Spray and Wait is comparable to that of Direct Delivery; it is always smaller, because the multiple copies of the message in the network, increase the probability of reaching the destination in less time.

PROPHET [21] implements the forwarding technique. When the number of nodes grows, the forwarding technique has more candidates as possible effective relays among the contacts. This implies, that the overhead increases (more relays imply more transmissions) and the delay decreases (more copies of the message increase the probability of reaching the destination). However, the effective management of the multiple copies of the message, requires either a proper queue management technique or a big buffer. In fact, for small size of the buffer, the delivery ratio of PROPHET decreases due to message dropping.

Fuzzy Spray [14] implements the queue management technique. It is the best performer in terms of delay, because (similarly to epidemic) there is unlimited number of replicas in the network. QM tries to keep messages in the queue until there is an high confidence that they have been already delivered.

As a consequence, even when the size of the queue is relatively small, the effects of buffer overflow and message dropping are less dramatic (note the differences with Epidemic). Indeed, delivery ratio is always remarkable high, except for the highest density of nodes (70 nodes in Figure 3); this is because, in such scenario, the QM policy employed by Fuzzy Spray do not effectively handle all the "important" messages.

Advanced Protocol.

The advanced protocols combine the proposed techniques. The effectiveness of such approach is shown in Figure 4 where the delivery ratio is always remarkably high. Furthermore, the overhead grows almost linearly (except for FAD) and it is in any case an order of magnitude smaller than that shown by single technique protocols. This confirms that, the combination of the proposed techniques can significantly increase the scalability of the protocols.

(a)(b)(c)

Figure 2: All with RW: (a)Delivery Ratio , (b)Delay, (c)Overhead.

(a)(b)(c)

Figure 3: All with RW: (a)Delivery Ratio , (b)Delay, (c)Overhead.

FAD [36] implements the queue management and the forwarding techniques. As observed before, its overhead is by far the highest in the considered scenario; it is an unexpected behaviour, because FAD is designed to limit the forwarding which is the main responsible for overhead. This can be explained recalling that a) in the considered scenario TTL is set to infinity and b) FAD keeps in queue the most important messages, namely the messages with lower fault tolerance values. Those messages are never evicted (TTL is infinity) and are often retransmitted increasing the overhead. To confirm this fact, we ran some other experiments with limited TTL, and indeed in this scenario, FAD shows a low overhead.

Also MaxProp [1] and RAPID [3] implement the queue management and the forwarding techniques and their performance are very similar. With respect to FAD, the main difference is in the forwarding policy which doesn't impose filtering on messages but only re-ordering. When a filter is applied, messages can be forwarded or not, while re-ordering implies that all messages can be possibly forwarded according to the new ordering, if there is a sufficient amount of time. The QM technique guarantees high delivery ratio (even with small buffer size) and low delay as well as limited overhead even if unlimited replication is employed.

SCAR [27] is the only protocols implementing all the proposed techniques. However, the SCAR queue management is very simple (it only uses master-backup copies) and in the considered scenario, the context information (i.e. battery level, mobility pattern etc.) are similar for all nodes limiting the effectiveness of the forwarding. As a consequence, SCAR shows high delay and relatively small delivery ratio, while the overhead is always small thanks to the effectiveness of the replication technique.

Mobility Models Impact.

Mobility models clearly affect nodes' opportunity to meet other nodes, namely the probability to reach the destination and the inter-contact probability. Consequently, also messages likelihood to be delivered before being dropped depends on mobility models.

In this section we report on some experimental activities performed to evaluate the impact of the mobility models on some of the protocols we discussed in the previous sections. Apart from Random Walk that is the mobility model used in the previous results, we consider three further mobility models briefly introduced in the following.
Random Waypoint(RWP)[18]. In RWP, nodes randomly choose the next destination and move there with a random speed. Upon arrival at the destination, the node pauses for a while and then chooses a new destination. *Levy Walk(LW)[29].* This mobility model tries to emulate human walk patterns in outdoor environments. The next direction is uniformly selected at random, while distance from the current position and speed are selected according to a power law distribution. *Community(CM).* In CM, the field in which nodes move is divided in areas called communities, and each node has a probability to get in/out from such areas as well as transit from on community to another [10]. Our implementation follows the specification given in [36].

We observe that, also the relevance of the information exchanged between nodes to update their knowledge on the environment in many cases strongly depends on mobility models. As an example, social information in RWP are useless while the same information in the CM is used to predict the expected movement pattern of nodes.

Figures 5 and 6 show the effects of different mobility models on the delivery ratio and delay of some of the protocols discussed earlier. When the movement pattern is controlled by some intentional behaviour (as in LW and CM), protocols using the knowledge (e.g. MaxProp) perform better and in any case the gap between such protocols and the ones that do not use the knowledge is remarkably high. When random movement patterns (as in RW, RWP) are considered, this gap becomes less evident, due to the fact that the knowledge acquired by nodes can be hardly used to predict future behaviours.

Surprisingly, the delivery ratio of knowledge based protocols is higher in the experiments based on random mobility models. As an example, the delivery ratio of MaxProp when 10 nodes are considered is more than 98% in figure 4, while is about 92% in figure 5.

This apparently strange behaviour can be explained by the fact that in the random mobility models, the probability of meeting other nodes is significantly higher and more uniform than in LW and CM. The evidence of this is that DirectDelivery protocol, which only delivers messages to the destination has a delivery ratio very high (more than 75% fro all densities) in the random mobility case (see figure 2) and relatively smaller for LW an CM (see figure 5). This fact ease the delivery of the messages to random selected destinations. In any case, we stress again that the gap between knowledge based protocols and other protocols is pretty evident when LW and CM are used.

Figure 5: Delivery ratio with different mobility models, 40 nodes

Figure 6: Delay with different mobility models, 40 nodes

5. CONCLUSIONS AND FUTURE WORK

(a)(b)(c)

Figure 4: All with RW: (a)Delivery Ratio , (b)Delay, (c)Overhead.

We identified forwarding, queue management and replication as the main techniques characterising DTN routing protocols. On the basis of such techniques, we propose a taxonomy and we classified many DTN routing protocols in the literature according to such taxonomy. Contrary to most previous papers, where either only qualitative comparisons have been presented or only a single category of protocols have been analysed, in our work we discuss the results of our experimental activity on many of the DTN protocols in the literature. Our results show that, an effective combination of the proposed techniques, can significantly improve the performance of the protocols in terms of delivery ratio, overhead and delay.

In this paper, we focus on one-to-one communications, but there are other communication patterns in which DTNs can be useful. As an example, DTNs can be used to re-connect Wireless Sensor Networks (WSNs) otherwise disconnected. We also plan to extend our simulations to other protocols and to consider real traces of mobile nodes, such the ones collected in a recent experiment conducted in our department [6]. Finally, we are finalising a prototypical implementation of an Android application capable of running some of the DTN protocols discussed in this paper.

6. REFERENCES

[1] Maxprop: Routing for vehicle-based disruption-tolerant networks. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1 – 11, April 2006.

[2] Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges. pages 24–37, 2006.

[3] *DTN routing as a resource allocation problem*, New York, NY, USA, 2007. ACM.

[4] *Routing Protocols in Delay Tolerant Networks: A Comparative Survey*, 2008.

[5] S. Ali, J. Qadir, and A. Baig. Routing protocols in delay tolerant networks - a survey. In *Emerging Technologies (ICET), 2010 6th International Conference on*, pages 70 –75, 2010.

[6] Lorenzo Bergamini. Soci@al-dis experiment. `https://sites.google.com/site/sapienzasocialexperiment/`.

[7] Chiara Boldrini, Marco Conti, Jacopo Jacopini, and Andrea Passarella. Hibop: a history based routing protocol for opportunistic networks. *A World of Wireless, Mobile and Multimedia Networks, International Symposium on*, pages 1 –12, june 2007.

[8] A.; Torgerson L.; Fall K.; Cerf V.; Durst B.; Scott K.; Weiss H.; Burleigh, S.; Hooke. Delay-tolerant networking: an approach to interplanetary internet. *Communications Magazine, IEEE*, pages 128 – 136, June 2003.

[9] Elizabeth M. Daly and Mads Haahr. Social network analysis for routing in disconnected delay-tolerant manets. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pages 32–40. ACM, 2007.

[10] Ha Dang and Hongyi Wu. Clustering and cluster-based routing protocol for delay-tolerant mobile networks. *Wireless Communications, IEEE Transactions on*, pages 1874 – 1881, june 2010.

[11] Etienne C. R. de Oliveira and Célio V. N. de Albuquerque. Nectar: a dtn routing protocol based on neighborhood contact history. In *Proceedings of the 2009 ACM symposium on Applied Computing*, pages 40 – 46. ACM, 2009.

[12] Henri Dubois-Ferriere, Matthias Grossglauser, and Martin Vetterli. Age matters: efficient route discovery in mobile ad hoc networks using encounter ages. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 257–266. ACM, 2003.

[13] Kevin Fall. A delay-tolerant network architecture for challenged internets. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 27 – 34. ACM, 2003.

[14] Mathurapoj A.; Pornavalai C.; Chakraborty G.;. Fuzzy-spray: Efficient routing in delay tolerant ad-hoc network based on fuzzy decision mechanism. In *Fuzzy Systems, 2009. FUZZ-IEEE 2009. IEEE International Conference on*, pages 104 – 109, Aug 2009.

[15] Radu Handorean, Christopher Gill, and Gruia-Catalin Roman. Accommodating transient connectivity in ad hoc and mobile settings. In *Pervasive Computing*, Lecture Notes in Computer Science, pages 305 – 322. Springer Berlin / Heidelberg, 2004.

[16] Pan Hui, Jon Crowcroft, and Eiko Yoneki. Bubble rap: social-based forwarding in delay tolerant networks. In *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, pages 241–250. ACM, 2008.

[17] Sushant Jain, Kevin Fall, and Rabin Patra. Routing in a delay tolerant network. In *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 145 – 158. ACM, 2004.

[18] Ari Keränen, Jörg Ott, and Teemu Kärkkäinen. The ONE Simulator for DTN Protocol Evaluation. In *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.

[19] Sotirios-Angelos Lenas, Stylianos Dimitriou, Fani Tsapeli, and Vassilis Tsaoussidis. Queue-management architecture for delay tolerant networking. In *Wired/Wireless Internet Communications*, pages 470 – 482. Springer Berlin / Heidelberg, 2011.

[20] A. Lindgren and K.S. Phanse. Evaluation of queueing policies and forwarding strategies for routing in intermittently connected networks. In *Communication System Software and Middleware, 2006. Comsware*

2006. First International Conference on, pages 1 – 10, 2006.

[21] Anders Lindgren, Avri Doria, and Olov Schelén. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, pages 19–20, 2003.

[22] L. Liu and Z. Chen, editors. *Data forwarding in opportunistic networks*. Information Technology Journal, 2010.

[23] Mengjuan Liu, Yan Yang, and Zhiguang Qin. A survey of routing protocols and simulations in delay-tolerant networks. In *Wireless Algorithms, Systems, and Applications*, pages 243–253. Springer Berlin / Heidelberg, 2011.

[24] R.S. Mangrulkar and M. Atique. Routing protocol for delay tolerant network: A survey and comparison. In *Communication Control and Computing Technologies (ICCCCT), 2010 IEEE International Conference on*, pages 210 –215, Oct. 2010.

[25] Shashidhar Merugu, Mostafa Ammar, and Ellen Zegura. Routing in space and time in networks with predictable mobility. Cc technical report; git-cc-04-07, Georgia Institute of Technology, 2004.

[26] Paulo Moreira, Waldir; Mendes. Survey on opportunistic routing for delay/disruption tolerant networks. Technical report, SITI Technical Report SITI-TR-11-02, 2011.

[27] Bence Pasztor, Mirco Musolesi, and Cecilia Mascolo. Opportunistic mobile sensor data collection with scar. *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference*, pages 1 – 12, 2007.

[28] A.; Conti M.; Pelusi, L.; Passarella, editor. *Opportunistic networking: data forwarding in disconnected mobile ad hoc networks*. Communications Magazine, IEEE, November 2006.

[29] Injong Rhee, Minsu Shin, Seongik Hong, Kyunghan Lee, Seong Joon Kim, and Song Chong. On the levy-walk nature of human mobility. *Networking, IEEE/ACM Transactions on*, pages 924 – 932, 2007.

[30] Gabriel Sandulescu and Simin Nadjm-Tehrani. Opportunistic dtn routing with window-aware adaptive replication. In *Proceedings of the 4th Asian Conference on Internet Engineering*, pages 103–112. ACM, 2008.

[31] Hemal Shah and Yogeshwar P. Kosta. Evolution of various controlled replication routing schemes for opportunistic networks. In *Advanced Computing*, pages 338–347. Springer Berlin Heidelberg, 2011.

[32] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In C. S. Raghavendra T. Spyropoulos, K. Psounis, editor, *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 252–259. ACM, Aug 2005.

[33] Thrasyvoulos Spyropoulos, Rao Naveed Rais, Thierry Turletti, Katia Obraczka, and Athanasios Vasilakos. Routing for disruption tolerant networks: taxonomy and design. *Wirel. Netw.*, pages 2349 – 2370, Nov. 2010.

[34] Kun Tan, Qian Zhang, and Wenwu Zhu. Shortest path routing in partially connected ad hoc networks. In *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, pages 1038 – 1042 Vol.2, 2003.

[35] Amin Vahdat and David Becker. Epidemic routing for partially-connected ad hoc networks. Technical report, Duke Technical Report CS-2000-06, 2000.

[36] Yu Wang and Hongyi Wu. Delay/fault-tolerant mobile sensor network (dft-msn): A new paradigm for pervasive information gathering. *IEEE Transactions on Mobile Computing*, pages 1021–1034, June 2007.

[37] Jeff Wilson. Probabilistic routing in delay tolerant networks. *Proceedings of 6th ICPCA 2011*, (November), 2007.

[38] Yi Xian, Chin-Tser Huang, and Jorge Cobb. Look-ahead routing and message scheduling in delay-tolerant networks. *Computer Communications*, pages 2184 – 2194, 2011.

[39] Quan Yuan, Ionut Cardei, and Jie Wu. Predict and relay: an efficient routing in disruption-tolerant networks. In *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '09, pages 95–104. ACM, 2009.