Selfish and Malicious Behavior in Delay-Tolerant Networks

Naércio Magaia, Paulo Rogério Pereira, Miguel P. Correia
INESC-ID/IST/UTL, Rua Alves Redol, 9. 1000-029 LISBOA, Portugal
{naercio.magaia, miguel.p.correia}@ist.utl.pt, prbp@inesc.pt

Abstract: Delay-Tolerant Networks (DTNs) are composed of nodes that cooperate to forward messages despite connectivity issues. This paper focuses on the problem of some nodes making limited or no contribution to the network. Misbehaving nodes consume network resources, reducing its performance and availability, therefore they constitute an important problem that should be considered. We study the impact of node misbehavior on seven DTN routing protocols using a large set of simulations. The results show that different protocols are more resilient to different types of node misbehavior.

Keywords: Delay-Tolerant Networks, Routing, Cooperation, Misbehavior.

1. Introduction

Delay Tolerant Networks (DTNs) [1] are composed of nodes that cooperate with each other to forward data despite connectivity issues, e.g., long and variable delays, high error rates, and intermittent connectivity. Due to their characteristics, DTNs are not amenable to traditional routing protocols for Mobile Ad-Hoc Networks (MANETs), like AODV. An interesting case is Vehicular Delay Tolerant Networks (VDTNs) [2], in which vehicles communicate wirelessly with each other on a DTN manner to disseminate messages. Some potential applications are notification of traffic conditions, weather reports, advertisements, and web or email access.

DTN routing protocols use a store, carry and forward approach, which implies some degree of cooperation among nodes, as nodes route other nodes' messages, or pick them in one place and deliver them in another. In order to overcome the lack of end-to-end paths, the protocols replicate messages, if necessary, in each contact.

The presence of misbehaving nodes, i.e., of nodes that do not follow the protocol –e.g., by using more than their share of resources – is a problem that has already been identified, but not thoroughly studied. One paper has shown that the performance of a DTN can be severely degraded if such nodes exist [3]. Nevertheless, misbehaving nodes are a real possibility. An important cause for misbehavior is selfishness: disseminating bogus delivery probability values in order to increase or decrease the probability of being chosen [3][4] or to save its own resources (storage, CPU, energy, etc.) [5]. Moreover, malicious attacks are pandemic in the Internet so they can also affect DTNs.

The contribution of this paper is a study of the impact of misbehaving nodes on several representative DTN routing protocols in terms of the number of copies created – single-copy, *n*-copy, and unlimited-copy – and if an estimation metric is used – estimation-based routing protocols. We evaluated the DTN routing protocols' performance in terms of delivery ratio, buffer time, hop count, latency and overhead ratio. Our work allows an adequate selection of DTN routing protocol if the presence of misbehaving nodes is possible or expected. However, we also conclude that there is no one-size-fits-all protocol, as different protocols perform better or worse depending on the type of misbehavior. This suggests the need for further research in the area.

The paper is organized as follows. Section 2 presents the routing protocols considered in the paper and discusses related work. Section 3 presents the types of misbehavior studied. Section 4 describes the simulation model. Section 5 presents the evaluation of DTN routing protocols with a variable number of misbehaving nodes. Section 6 concludes the paper.

2. DTN Routing Protocols and Related Work

Direct Delivery [6] and First Contact [7] are single copy DTN routing protocols where only one copy of each message exists in the network. In Direct Delivery, the message is kept in the source and delivered only to the final destination, if the nodes meet. In First Contact, the message is forwarded to the first node encountered and deleted. The message is forwarded until it reaches the intended destination.

The Epidemic [8] routing protocol is an unlimited-copy routing protocol as nodes may forward messages to any node they come in contact with. When two nodes come into communication range, they exchange a summary vector containing information about messages that they have not yet seen. The receiving node decides whether it accepts the message or not. This decision is taken, for example by not carrying messages for a certain destination node, or of a certain size. The buffer size and hop count field limit the amount of resources consumed through Epidemic Routing.

In many real environments, encounters between nodes are not random, but follow a predictable pattern. Mobile Ubiquitous LAN Extensions (MULEs) [9] are mobile agents (vehicles or animals) that when in close range pick, buffer and drop off data, carrying data between remote locations. To exemplify a pattern, the Data MULEs in [10], meet with higher probability certain Data MULEs. The Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET) protocol [11] uses a probabilistic metric: delivery predictability, that attempts to estimate, based on node encounter history, which node has the higher probability of successful delivery of a message to the final destination. When two nodes are in communication range, a new message copy is transferred only if the other node has a better probability of delivering it to the destination.

MaxProp [12] attempts to transfer all messages not held by the other node, when it is in communication range. The protocol uses acknowledgments to clear the remaining copies of a message in the network when it is received by the destination node. When nodes discover each other, MaxProp exchanges messages in a specific priority order, taking into account message hop counts and the delivery likelihood to a destination based on previous encounters. New packets are assigned higher priority, and the protocol attempts to avoid reception of duplicate packets.

In the Resource Allocation Protocol for Intentional DTN (RAPID) [13], routing packets are opportunistically replicated until a copy reaches the destination node. The protocol models DTN routing as a utility-driven resource allocation problem. The routing metric is a per-packet utility function. When nodes are in communication range, RAPID replicates the packet that results locally in the highest increase in utility. The corresponding utility U_i of packet *i*, is defined as the expected contribution of *i* to the given utility routing metric. RAPID is composed of three core components: (1) a selection algorithm determines which packets to replicate given their utilities when nodes are in communication range, (2) the inference algorithm, that given a routing metric estimates the utility of a packet, and (3) a control channel, that propagates metadata required by the inference algorithm.

Spray and Wait [14] is an *n*-copy routing protocol with two phases: (1) *spray phase*, where a message created by the source node is initially spread by the source to encountered nodes until the *n* copies are exhausted; (2) *wait phase*, where every node containing a copy of the message performs a direct delivery to the destination. There are two variants of the protocol: normal mode, where a node gives one copy of the message to each node it discovers that does not have the message; and binary mode, where half of the *n* copies are given in each encounter.

Table 1 summarizes the DTN routing protocols and their characteristics.

Table 1: DTN Routing Protocols

Routing Protocol	Abbreviations	*-copy	Estimation-based
Direct Delivery [6]	DD	single-copy	No
First Contact [7]	FC	single-copy	No
Epidemic [8]	Epidemic	unlimited-copy	No
PRoPHET [11]	Prophet	unlimited-copy	Yes
MaxProp [12]	Maxprop	unlimited-copy	Yes
RAPID [13]	Rapid	unlimited-copy	Yes
Spray and Wait [14]	SnWNormal / SnWBinary	*n*-copy	No

There is also some related work about node misbehavior in DTNs. The authors of [5] analyze a situation where network resources are not managed. These resources can be abused by individuals, called resource hogs, who attempt to send more of their own data and less from other nodes'. Consequently, less data from other nodes is delivered. The authors proposed a buffer management mechanism to protect honest users from resource hogs.

In [4], an encounter prediction system is proposed that is secure against malicious nodes that provide forged metrics to other nodes that come in contact with, in order to attract packets to them. Malicious nodes can either drop or utilize the received packets to launch more sophisticated attacks. The encounter predication system consists of history interpretation, competency evaluation, evidence sufficiency checking, and aging. Nodes using this system make forwarding decisions which prevent attackers from boosting their routing metrics.

The authors in [3] use the concept of reputation to address the case of misbehaving carriers, as DTN performance and availability degrades when nodes and carriers do not cooperate with each other. Misbehaving carriers may increase or decrease this probability of being chosen. Reputation is measured as the trustworthiness of carriers. Low values of reputation correspond to misbehaving carriers, whereas high values correspond to well-behaving carriers. By using this mechanism, a misbehaving carrier is not chosen even though it disseminates forged values of delivery probability.

In [15] and [16], studies of the effects of cooperation in DTNs are presented. The authors in both studies considered three routing protocols and studied the performance of these routing protocols in a non-cooperative environment, in terms of delivery delay and transmission overhead [15] and in terms of message delivery performance [16]. Both works are close to ours, but we have considered more routing protocols, more metrics, and different types of misbehavior.

3. Misbehavior Types

A DTN relies on the fact that nodes cooperate with each other to forward messages [15][16]. A node can misbehave, i.e., to fail maliciously, generically in two ways [17]: (1) by doing content failures, i.e., delivering modified messages; and (2) by doing timing failures, i.e., delivering messages out of time (or not at all, which corresponds to infinite delay, or repeatedly).

We do not consider the first class of misbehavior because it is simple to translate into omissions, which are timing failures. The mechanism to make this translation consists simply in the sender concatenating a digital signature (obtained, e.g., with RSA or DSA) or an MAC (obtained, e.g., with HMAC-SHA1) [18] to every message sent; the receiver verifying the signature/MAC and discarding the message if the verification fails. This mechanism guarantees that if a node corrupts a message, this is equivalent to discarding it, except for the resources consumed. Alternatively, using digital signatures it is also possible to have any non-malicious node doing the verification (using the public-key) and discarding corrupted messages.

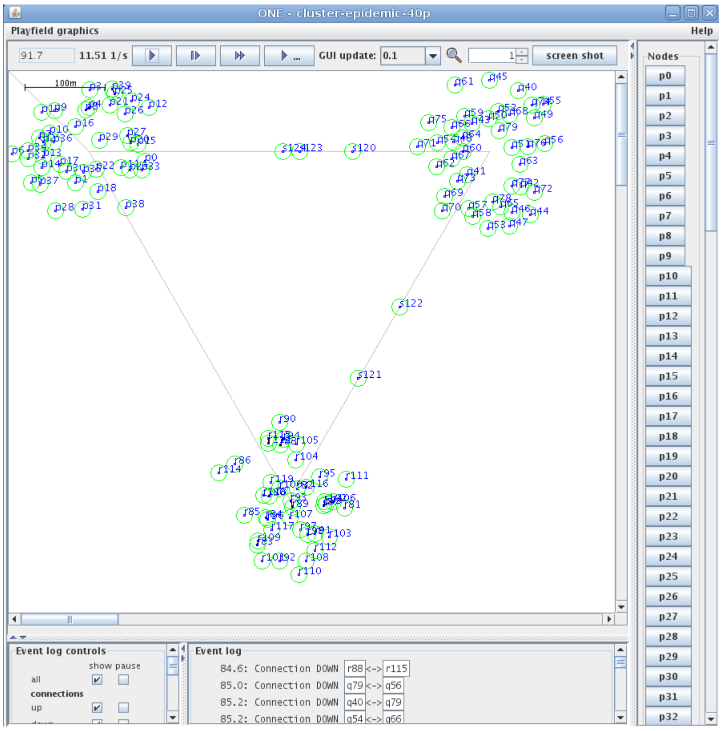
We considered two types of misbehavior:

- **Type I**. Upon message reception, the node drops it. It is the quintessential selfish behavior.
- **Type II**. Upon message reception, the node drops it. Upon message creation, the node creates ten additional new messages and sends them. This behavior is inspired in the malicious resource hogs of [5].

Despite these forms of misbehaviors, we assume that all nodes receive the messages destined for them.

4. Simulation Model

We used the ONE Simulator [7] on an example scenario, see Figure 1, consisting of a network with 125 nodes: 120 pedestrians and 5 trams. Our scenario resembles those in [10][19]. We choose a simulation time of 24h with an update interval[1] of 0.1 s. We modeled the malicious behavior as described on Section 3, and examined their effect on the 8 routing protocols of Table 1. We considered 6 copies for Spray and Wait. We considered from 20% to 80% misbehaving nodes over the total number of pedestrian nodes, in intervals of 20%. We did not consider misbehaving trams.



ONE - cluster-epidemic-40p

GUI update: 0.1

Event log controls

	show	pause
all	☑	□
connections		
up	☑	□

Event log

```
84.6: Connection DOWN r88<->r115
85.0: Connection DOWN q79<->q56
85.2: Connection DOWN q40<->q79
85.2: Connection DOWN q54<->q66
```

Figure 1 Our example scenario on the ONE simulator's GUI

Mobility. We used a cluster based mobility model with three clusters (each cluster can be a remote village) over an area of 4.5 × 3.4 Km. We used trams (a tram can be a message ferry [1]) to connect the clusters. Inside of each cluster, the pedestrians were moving in a

[1] Update interval is defined as the time step increment for the simulation time.

speed varying between 0.5 to 1.5 m/s and between the cluster, the trams were moving at a speed varying from 3 to 5 m/s. Each time a tram reaches its destination, it pauses for a time varying from 10 to 30 seconds.

Connectivity and transmission. Only two nodes can communicate with each other at a time, within range. The communication range between the nodes is of 10 m, and the communication is bi-directional at a constant transmission rate of 250 kB/s.

Traffic model. Every 5 to 10 minutes, a source node randomly chosen can generate one message to a randomly chosen destination. Trams do not generate messages, being only used to carry messages between clusters. Nodes do not change their behavior (malicious or not) over time. The time-to-live (TTL) attribute of each message is 5h, and the message size varies from 100 kB to 2 MB.

Buffer management. The pedestrian and tram nodes have buffers for DTN traffic of 20 MB and 100 MB, respectively.

5. Simulation Results

We evaluate the performance according to the following metrics: delivery ratio, buffer time, hop count, latency and overhead ratio. The delivery probability is a key performance indicator of the simulation, as it tells the percentage of successfully received packets of all sent. Buffer time indicates for how long the messages were queued in the node's buffers. Hop count indicates the number of nodes the packet traversed with the exception of the source node. Latency is the time for a successful message delivery. Overhead is the number of message transmissions for each created message.

We present the values in the graphs with 95% confidence intervals. For cases where there are large differences in values, a logarithmic scale is used in the ordinate axis.

We ran thirty independent simulations using different seeds for each protocol-percentage pairs, and the results were averaged. Simulations usually run much faster than in real-time. We observed mean simulation speeds ranging from 90:1 to 300:1 (ranging from 3 to 15min per simulation), depending on the routing protocol and the percentage of misbehaving nodes; only Rapid was slower (as low as 30:1 and less), taking almost 9h per simulation.

5.1 – Average Message Delivery Probability

Figure 2 shows the average message delivery probability for all protocols with the two types of misbehavior. DD is not affected by Type I misbehavior as nodes only deliver messages when they meet the message recipient. An issue that affects DD's delivery probability is the cluster scenario, since communication between clusters is made through trams. So, DD delivery probability must be below 1/3, as nodes in different clusters never meet directly.

Spray and Wait is similar to DD in that during the wait phase it performs direct transmissions. Because of the spray phase, some messages generated to nodes in another cluster are delivered, which allows the protocol to have a delivery probability above 1/3.

First Contact forwards only one copy of a message to the first node met. Because of this, it is the DTN routing protocol most affected by misbehaving nodes, as even if the network only contains 20% of misbehaving nodes, the probability of meeting a misbehaving node that drops the message forever is high, resulting in a reduction of 94% of the delivery probability.

Misbehaving nodes cause a reduction of the number of message copies circulating in the network (congestion) as they drop them. Protocols like Epidemic and Prophet take advantage of small percentages (below 80%) of Type I misbehaving nodes as their delivery probabilities increases with the reduction of network traffic. Maxprop and Rapid have the best delivery probabilities for Type I, since the replication and discarding mechanisms used by both ensure selection of the best sets of messages to transmit and/or remove, respectively. For Type II, all protocols experienced reductions in the delivery probability as misbehaving nodes degrade the network conditions.

Protocols, like MaxProp and RAPID, are barely affected by Type I misbehaving nodes. But due to the additional overhead caused by Type II misbehaving nodes, their performance degrades with the increase of the percentage of misbehaving nodes.

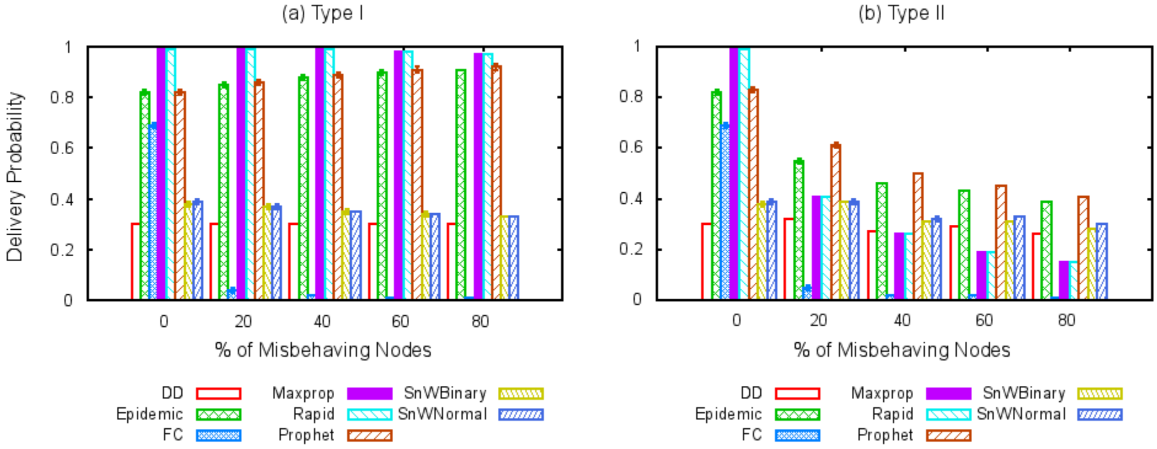


Figure 2 Average Delivery Probability as function of the percentage of Types I and II misbehaving nodes

5.2 – Average Message Buffer Time

Due to the direct transmissions approach used by DD and Spray and Wait, they present the highest values of buffer time in comparison with other protocols. For both types of misbehaving nodes, DD presents buffer time values very close to the maximum TTL, see Figure 3, as the protocol buffers messages until it finds the destination nodes or drops them if they expire. Nevertheless, Type II presents a slight reduction on the buffer time with the increase of misbehaving nodes. This happens because of the high number of messages generated by the Type II misbehaving nodes which cause an increased radio usage preventing other nodes in communication range from delivering their messages.

Both versions of Spray and Wait suffer more with misbehaving nodes due to the spray phase, as the source node can replicate messages to misbehaving nodes that silently drop them. As a consequence, with the increase of the percentage of misbehaving nodes, the protocol tends to reduce buffer time.

Other DTN routing protocols presented in this paper, despite their routing algorithms, have smaller values of average buffer time as they replicate messages to encountered nodes more often, which reduces buffer time.

5.3 – Average Message Hop Count

Figure 4 show that when nodes do not misbehave First Contact (FC) presents the highest hop count. This is due to the fact that FC forwards messages to the first encountered nodes and these messages are continuously forwarded until they reach the intended destination node. The increase of misbehaving nodes causes a reduction in the number of hops travelled by messages in all routing protocols, for the same reasons as it caused a reduction in the delivery probability.

As expected, DD has the smallest value of hop count due to the use of a direct transmission approach. Because of the spray phase, Spray and Wait has a few more hops.

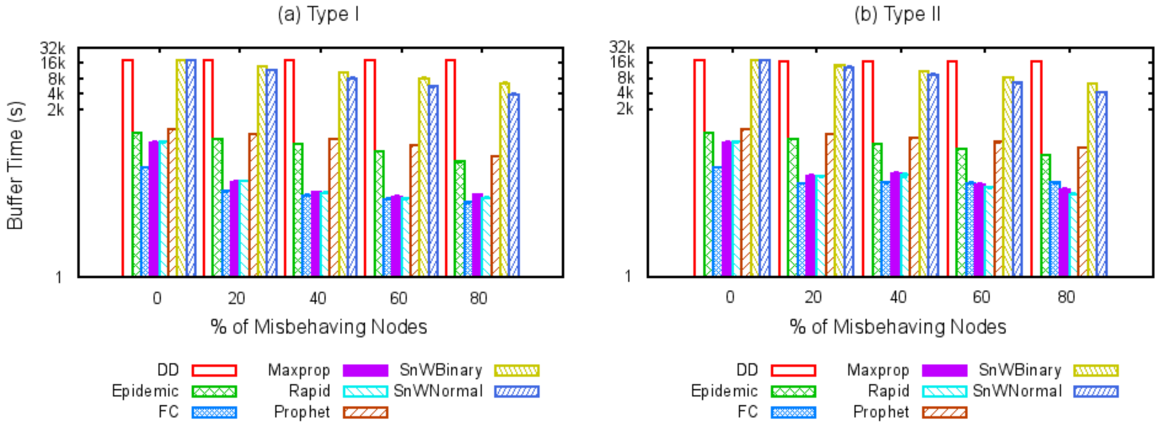


Figure 3 Average Buffer Time as function of the percentage of Types I and II misbehaving nodes

In general, the increase of misbehaving nodes causes a reduction in the hop count as nodes can only rely on the remaining amount of well-behaved nodes to store, carry and forward messages. As the percentage of well-behaved nodes reduces, only the messages that travel fewer hops (closer to one) are delivered, unless the protocol has a mechanism of only selecting a well-behaved node as forwarder.

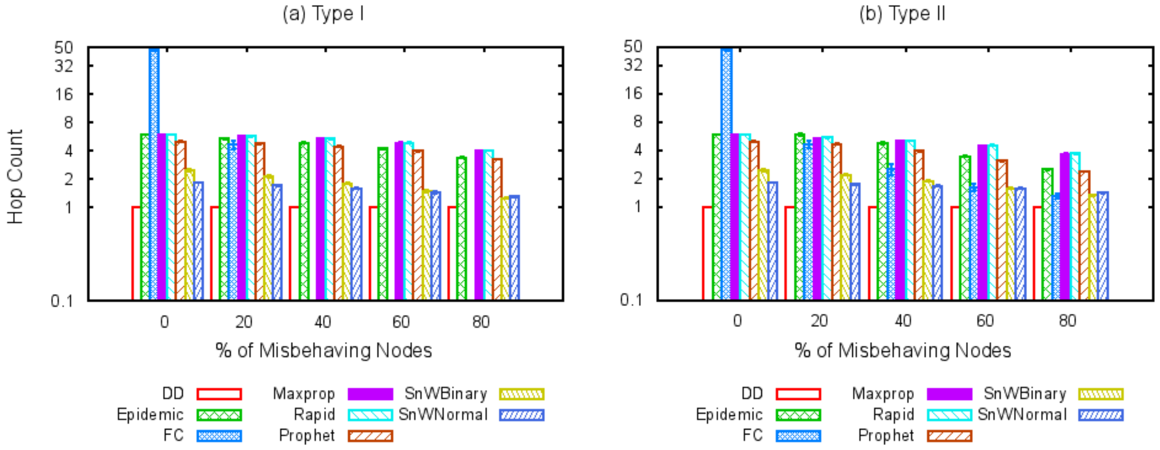


Figure 4 Average Hop Count as function of the percentage of Types I and II misbehaving nodes

5.4 – Average Message Latency

For scenarios were nodes do not misbehave, FC also presents high values of latency due to the high number of hops, as can be seen in Figure 5. FC is also the only protocol in which there is a reduction of latency with the increase of misbehaving nodes. This happens because undelivered messages do not contribute to the latency statistics and fewer messages are delivered as more misbehaving nodes drop them.

For Type I misbehaving nodes, DD’s latency does not change. The average DD latency decreased for Type II misbehaving cases, as the protocol is slightly affected by the radio usage to transfer additional messages, generated by misbehaving nodes.

For all types of misbehaving nodes, Maxprop and Rapid’s latencies increased with the increase of the percentage misbehaving nodes. Since messages travel on average a similar number of hops, the increase of misbehaving nodes caused an increase in network latency. This happens because messages with small latency values are dropped more and more by the increasing number of misbehaving nodes. Messages with small hop count tend to have smaller latency values, being more commonly dropped by misbehaving nodes.

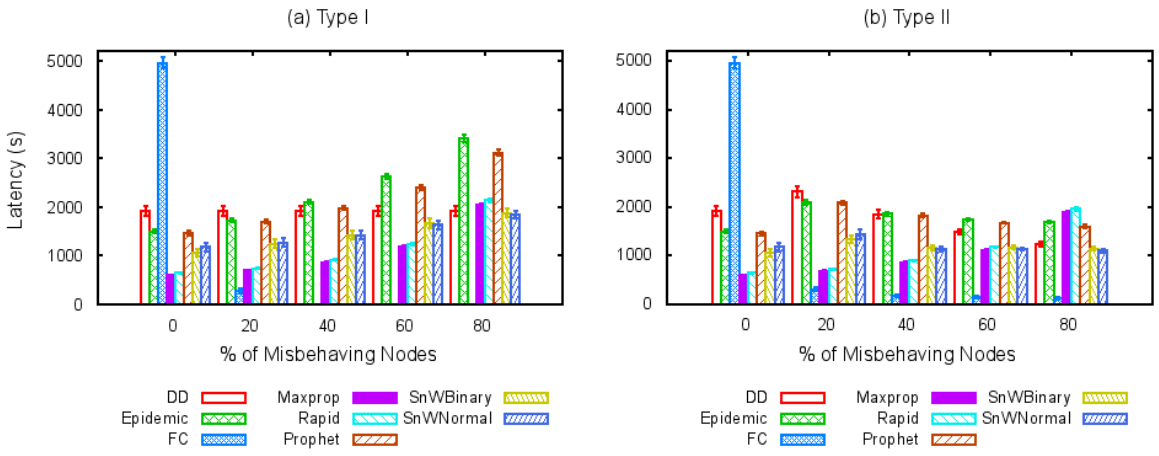


Figure 5 Average Latency as function of the percentage of Types I and II misbehaving nodes

5.5 – Average Message Overhead Ratio

Figure 6 shows that Epidemic and Prophet have the highest values of overhead ratio. This happens because of the similarities between these DTN routing protocols. Prophet only has smaller overhead because it uses a probabilistic metric that decides if it is worth replicating a message to a contacted node.

Due to the similarities between DD and both versions of Spray and Wait, DD has zero overhead and Spray and Wait has smaller values of overhead ratio in comparison with other DTN routing protocols.

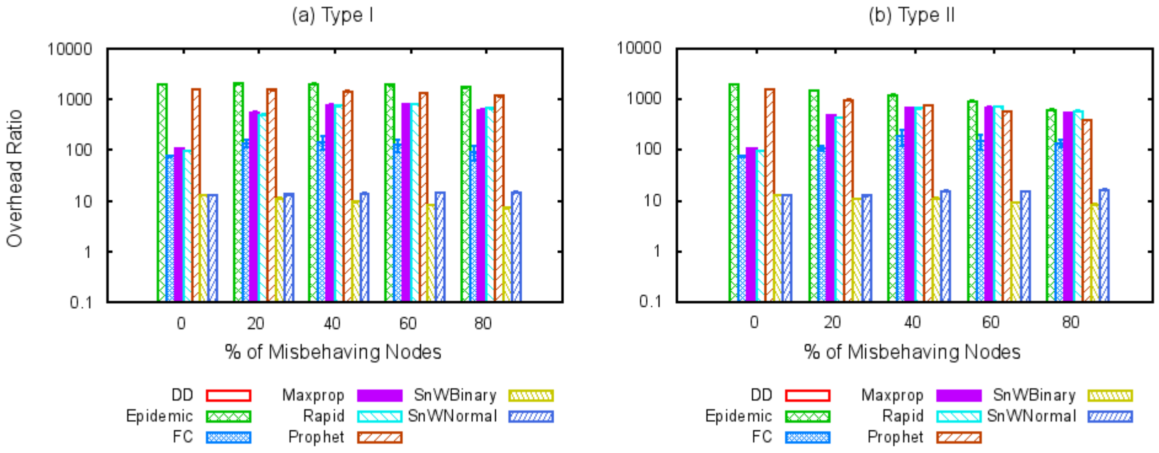


Figure 6 Average Overhead Ratio as function of the percentage of Types I and II misbehaving nodes

6. Discussion and Conclusion

We studied the impact of two types of node misbehavior on seven DTN routing protocols. The simulations show that in the presence of misbehaving nodes, Epidemic and Prophet are quite robust as they have no limits to message replication and may even benefit from misbehaving nodes as they may reduce network congestion. Maxprop and Rapid, for scenarios without misbehaving nodes, have the highest values of delivery probability. But if we consider scenarios with misbehaving nodes, the protocols are strongly affected by these types of nodes. Both versions of Spray and Wait have low values of delivery probability because of the scenario used. For cases of source and destination nodes located in different clusters, unless the message was sprayed to a tram and then to a node inside the destinations’ node cluster, the message will never reach the destination, so the delivery probabilities are around 1/3, as for Direct Delivery. In presence of misbehaving nodes, First Contact is the most affected routing protocol, as there is a single message copy that being delivered to any malicious node is lost forever.

The main conclusion of this work is that the delivery probability (Section 5.1) of DTN routing protocols, in the presence of node misbehavior, depends on the type of misbehavior. With Type I misbehavior, Maxprop and Rapid provided the best results, followed by Prophet and Epidemic. With Type II misbehavior, Prophet and Epidemic were the best. Nevertheless, all these protocols were the worst in terms of overhead (Section 5.5). Direct Delivery, First Contact and the two variations of Spray and Wait had both poor performance. First Contact is, however, the routing protocol most affected by misbehavior, as it is also the one that is single-copy.

An interesting question is what characterizes protocol resilience to misbehavior. We classified the protocols in terms of two metrics, "*-copy" and "estimation-based" (cf. Table 1). In terms of the first, clearly the best protocols were the unlimited-copy ones' and the worst was First Contact that is a single-copy, with Spray and Wait (*n*-copy) in the middle. In relation to the second metric, estimation-based protocols are apparently better. Epidemic is not estimation-based and fares well, but it is also a brute-force protocol that does flooding, which therefore has the highest overhead.

As future work, we plan on evaluating the impact of misbehaving nodes in different conditions, like different types of nodes, different scenarios, and other types of node behaviors. It is interesting to test different scenarios, as it has influence on the nodes contacts pattern. Another issue that affects contact patterns are mobility models. Finally, we intend to propose new mechanisms to make routing more robust in the presence of misbehaving nodes.

Acknowledgments

This work was supported in part by national funding from FCT - Fundação para a Ciência e a Tecnologia (INESC-ID multiannual funding) through the PIDDAC program funds under project PEst-OE/EEI/LA0021/2011 and by the MPSat project (FCT Ref. PTDC/EEA-TEL/099074/2008).

References

[1] M. Khabbaz, C. Assi and W. Fawaz, "Disruption-Tolerant Networking: A Comprehensive Survey on Recent Developments and Persisting Challenges," *IEEE Communications Surveys & Tutorials*, 14(2):607-640, 2012.

[2] P. Pereira, A. Casaca, J. Rodrigues, V. Soares, J. Triay, C. Cervello-Pastor, "From Delay-Tolerant Networks to Vehicular Delay-Tolerant Networks," *IEEE Communications Surveys & Tutorials*, vol.14, no.4, pp.1166-1182, Fourth Quarter 2012.

[3] G. Dini and A. Lo Duca, "A reputation-based approach to tolerate misbehaving carriers in Delay Tolerant Networks," *In Proc. IEEE Symposium on Computers and Communications (ISCC)*, pp.772-777, Jun. 2010.

[4] F. Li, J. Wu and A. Srinivasan, "Thwarting Blackhole Attacks in Disruption-Tolerant Networks using Encounter Tickets," *In Proc. IEEE INFOCOM*, pp.2428-2436, Apr. 2009.

[5] J. Solis, N. Asokan, K. Kostiainen, P. Ginzboorg, and J. Ott., "Controlling resource hogs in mobile delay-tolerant networks," *Computer Communications*, vol. 33, pp. 2-10, Jan. 2010.

[6] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Single-copy routing in intermittently connected mobile networks," *In Proc. IEEE Secon'04*, 2004.

[7] A. Keränen, J. Ott and Teemu Kärkkäinen. "The ONE simulator for DTN protocol evaluation," *In Proc. 2nd International Conference on Simulation Tools and Techniques (Simutools)*, 2009.

[8] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," *Technical Report CS-200006*, Duke University, Apr. 2000.

[9] R. Shah, S. Roy, S. Jain and W. Brunette , "Data MULEs: Modeling a Three-tier Architecture for Sparse Sensor Networks," *Intel Research Tech Report IRS-TR-03-001*, Jan. 2003

[10] Eric Brewer, et. al., “The Case for Technology in Developing Regions,” *IEEE Computer*, vol. 38, no. 6, pp. 25-38, Jun. 2005

[11] A. Lindgren, A. Doria and O. Schelen, “Probabilistic routing in intermittently connected networks,” *In Proc. First International Workshop on Service Assurance with Partial and Intermittent Resources (SAPIR)*, 2004.

[12] J. Burgess, B. Gallagher, D. Jensen and B. Levine, “MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks,” *In Proc. IEEE INFOCOM*, Apr. 2006.

[13] A. Balasubramanian, B. Neil Levine and A. Venkataramani, “DTN routing as a resource allocation problem,” *In Proc. ACM SIGCOMM*, Aug. 2007.

[14] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks,” *In Proc. ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN)*, 2005.

[15] A. Panagakis, A. Vaios and I. Stavrakakis, “On the Effects of Cooperation in DTNs,” *In Proc. 2nd International Conference on Communication Systems Software and Middleware (COMSWARE)*, pp.1-6, Jan. 2007.

[16] A. Keranen, M. Pitkanen, M. Vuori and J. Ott, “Effect of non-cooperative nodes in mobile DTNs,” *In Proc. 2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp.1-7, Jun. 2011.

[17] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, “Basic Concepts and Taxonomy of Dependable and Secure Computing,” *IEEE Transactions on Dependable and Secure Computing*, vol 1, no. 1, pp. 11-33, Jan.-Mar. 2004.

[18] A. J. Menezes, p. C. van Oorschot and S. A. Vanstone, “Handbook of Applied Cryptography”. CRC Press, 1997.

[19] S. Jain, K. Fall and R. Patra, “Routing in a delay-tolerant network,” *In Proc. ACM SIGCOMM*, 2004.