On the Effects of Cooperation in DTNs

Antonis Panagakis, Athanasios Vaios and Ioannis Stavrakakis

Department of Informatics & Telecommunications
National & Kapodistrian University of Athens
Panepistimiopolis, Ilissia, 157-84, Athens, Greece
E-mail: {apan, avaios, ioannis}@di.uoa.gr

Abstract— **In a Delay Tolerant Network (DTN) the nodes may behave autonomously deciding on their own whether to implement or not the rules of a routing algorithm. In this paper, the effects of node cooperation (or lack of it) are explored for three well-known routing algorithms proposed for DTNs with respect to the message delivery delay and the transmission overhead incurred until message delivery or the termination of the message spreading process. The results show that the sensitivity of the algorithms to the cooperation degree can be high, to the point of making them inferior to algorithms they typically outperform under a fully cooperative environment. Finally, it is demonstrated how a simple mechanism that incorporates the cooperation degree can help improve effectiveness.**

Keywords: delay tolerant networks, routing, cooperation

I. INTRODUCTION

The Delay Tolerant Networking (DTN) paradigm, [1], is characterized by the lack of guaranteed connectivity and the typically low frequency of encounters between a given pair of nodes within the network. The routing algorithms proposed for DTNs rely on node mobility for message delivery and may be categorized into single- and multiple-copy algorithms depending on whether they allow the multiplication of the message within the network.

In single-copy strategies, there is only one copy of the message relayed among the nodes in the network until it is delivered to the destination, [2]; for instance, the message copy may be forwarded based on the maximization of a utility function at each node encounter, [3]. In multiple-copy strategies, the copies of the messages allowed to be spread may or may not be limited. In the case of a pure flooding mechanism, [4], all nodes simply exchange their copies upon encounter. When the copies of the message are limited, [5], a variety of message spreading algorithms has been proposed, including schemes in which the source is the only node allowed to relay to all others (and not only to the destination) and schemes in which nodes are allowed to share the (limited) number of copies they have with nodes they encounter until they reach the destination.

The performance of the various DTN routing algorithms proposed in the literature has been investigated so far with respect to the impact of the characteristics of the environment, e.g. the size of the area where the network is deployed and the node density, or the employed message spreading algorithm. Node behavior has been rarely considered, besides node mobility, and in all the studies it has been assumed that the nodes cooperate fully. The latter can be a fairly unrealistic assumption, as the participating nodes are autonomic, in the sense that they can decide on their own whether to implement or not the rules of a DTN routing algorithm. However, it is expected that the degree of node cooperation in a DTN would have a major impact on the performance of a DTN routing algorithm. It may be the case that if cooperation is not guaranteed, efficient DTN routing protocols almost collapse or yield a very poor performance while less efficient protocols are only marginally affected. The focus of this paper is to explore the impact of node cooperation on some representative routing algorithms for DTNs.

Cooperation has been studied under the framework of peer-to-peer and ad-hoc networks. The major issues typically considered are: the effect that cooperation might have on the network performance, the detection of non-cooperative behavior, and the design of mechanisms to enforce cooperation. Simple punishment mechanisms have been designed to address cooperation problems, based on game theoretic approaches aiming to provide incentives in order for the nodes to cooperate, [6], [7]. Reputation mechanisms have also been considered that monitor the nodes' past actions, keep a record of them, and then use this history to decide on packet forwarding or not, [8], [9]; this is actually a more complicated punishment method in which each node is handled and punished in a separate way and only based on its own previous actions. Another approach to deal with node cooperation is through the implementation of pricing mechanisms. In these mechanisms, credits are taken away from nodes that do not cooperate and credits are given to those that participate in packet forwarding, [10].

Here, we do not consider how such mechanisms could be applied to DTNs but we focus only on the performance of routing in a non-cooperative environment. Three representative routing algorithms are considered, ranging from a conservative scheme where only the source node is responsible for spreading the message copies within the network, to a fully-aggressive scheme

that floods the network with message copies. We study the performance of these algorithms in terms of the induced delivery delay and the transmission overhead; transmissions are considered not only until message delivery but also until the actual message spreading is ended. Cooperation is captured in terms of the node's probability to drop a message copy upon reception and/or to forward the message copy upon node encounter. By considering a simple strategy that takes into consideration the nodes' cooperation degree, it is demonstrated how one can alleviate the effects of non-cooperative behavior in DTNs.

II. MOBILITY-ASSISTED ROUTING IN DTNs

In our study, we use the following (three) multi-copy algorithms that cover a representative range of the relaying rules that may govern a routing strategy in a DTN.

- **Epidemic**: This algorithm is based on epidemic routing, as described in [4]. Every time two nodes encounter, they exchange their message copies. This algorithm provides the minimum message delivery delay but suffers from high buffer occupancy and high bandwidth utilization due to the large number of copies that are allowed to be spread within the network.

- **Two-Hop**: According to the two-hop relaying algorithm, [11], [12], the source is allowed to spread up to a maximum number of copies within the network. Each time it encounters some other node with no copy of the message, it gives it one until it has only one copy (for the destination node only). The intermediate nodes are not allowed to spread the message copy they may have to any other node than the destination.

- **Binary Spray and Wait**: According to this algorithm, [5], every node gives half of its message copies to every node with no copy it encounters until it has only one copy (to give it to the destination node). Binary spray and wait is faster than the two-hop relaying algorithm and induces no more transmissions than those of the latter.

Regarding the performance of a routing algorithm applied in DTNs there is a trade-off between the message delivery delay achieved and the overhead induced. The overhead has been measured in the literature as the number of transmissions spent until the message delivery; however, in most of the cases the actual overhead is more than that. This is because the message copy spreading process does not necessarily terminate when the message is delivered to the destination, but only when all the nodes participating in the copy spreading process are informed about the successful delivery (or, obviously, when there are no more copies to be spread). For this reason, we study the following overhead components:

- **Till Delivery**: It refers to the number of transmissions required until the message is delivered to the destination.

- **Additional**: It refers to the number of transmissions made after the message is delivered to the destination.

- **Total**: It refers to the total number of transmissions and is equal to the sum of the above overheads.

As far as the additional overhead is concerned, it is assumed that a notification procedure is activated upon the message delivery aiming at informing all the involved in the spreading process nodes of the delivery success.

Under epidemic routing, all the nodes that possess a message copy and are not yet notified of the message delivery will generate an additional transmission upon encountering a node that is also unaware of the successful message delivery. A node that has become aware of the message delivery is referred to as a *notifier*; clearly, the first notifiers are the nodes who deliver the message to the destination and the destination upon receiving the message. When a node that possesses a copy of the message encounters a notifier, this node drops the message and becomes a notifier as well. Under the binary spray and wait algorithm, the notification process is similar to that described above under epidemic routing; it should be noted, though, that the spreading process is ended anyway (independently of the notification process) when the maximum allowed number of copies are already spread. Under the two-hop relaying algorithm, additional overhead may be induced only after an intermediate node has delivered the message to the destination and the source node has not yet spread the maximum number of copies allowed to be spread; under the latter conditions, the source will continue spreading copies to other intermediate nodes until it is notified by a notifier.

III. COOPERATION EFFECTS

Cooperation in this paper refers to the node's willingness or ability to participate in the message spreading process; a node may be willing but unable to cooperate due to resource (buffer, energy, etc.) constraints. More specifically, a node may either drop a message upon reception or keep the message but avoid some of the forwarding transmissions that it is supposed to make according to the routing algorithm. In view of the above, the following two types of node behavior (cooperation) are considered here:

- **Type I**: Upon reception of the message copy, the node either drops it with probability P_{drop} or keeps it and follows the rules of the routing algorithm. This type of cooperation may be simply the result of node misbehavior or the node's inability to store the message due to buffer limitations. For this type of cooperation, the complementary probability (1-P_{drop}) will be referred to as the cooperation degree.

- **Type II**: Upon reception of the message copy, the node maintains the copy in its buffer but forwards it with a probability $P_{forward}$ that is typically less than 1. This type of cooperation may be simply the result of node misbehavior (less aggressive than in Type I case) or the node's inability

to afford the energy for all the needed transmissions due to energy constraints. For this type of cooperation, $P_{forward}$ will be referred to as the cooperation degree.

The lack of cooperation among the nodes may not be necessarily the result of a selfish behavior, but rather of a common strategy agreed and followed by all nodes to cater to their restricted capabilities and protect their limited resources. For instance, consider the case of a network with nodes of different energy levels. If E_i denotes the energy level of node i and node i forwards a message with a probability P_i that is proportional to its energy level ($P_i \sim E_i$), then the energy consumption due to forwarding is expected to be roughly proportional to the energy available at each node. This way, for example, the lifetime of all the nodes is expected to be roughly the same, independently of their initial energy supply.

In order to quantitatively characterize the sensitivity of each routing algorithm with respect to the degree of cooperation, we use two different metrics. The first one results from the comparison of the algorithm's performance for a certain degree of cooperation with that achieved in a fully cooperative environment; the greater the difference in the algorithm's performance in these two cases is, the more sensitive to cooperation the algorithm may be characterized. The second sensitivity metric is defined for Type I cooperation and is based on the comparison of the algorithm's performance for a certain degree of cooperation with that achieved in the *Fully Cooperative Equivalent (FCE)* network that is defined as follows: The FCE of a network of N nodes, each of which has a cooperation degree of $1-P_{drop}$ is defined as a network of N´ fully cooperative nodes, where N´=N $(1-P_{drop})$. The definition of the FCE network and its use for measuring an algorithm's sensitivity to cooperation is based on the fact that N´ is the expected number of nodes that would not drop a message copy in the original network and, thus, may be considered as the effective number of fully cooperative nodes that are present in the network at each time instant. The more the performance in the original network deviates from that in its FCE, the more sensitive an algorithm may be considered to be. Both sensitivity metrics are calculated for the mean delivery delay and the number of transmissions that take place.

We simulated a network of up to 100 nodes uniformly distributed in an area of 8km x 8km. Each node moves within the area, according to the Random Direction Model, [13], with a speed of 3m/s. When a copy is transmitted, a receiver can receive the data correctly when it is as far as 200m away from the sender. All the results are averaged values over 10000 runs.

A. Non-cooperative environments

In figures 1 and 2, the performance of the three algorithms is illustrated (both in terms of the achieved mean delay and the total overhead induced, respectively) as a function of the cooperation degree (for four different values 1, 0.75, 0.5 and 0.25 of Type I), for the case of 100 nodes (N) and a maximum number of message copies (K) equal to 100. In Fig. 3, the distribution of the overhead into its components is illustrated.
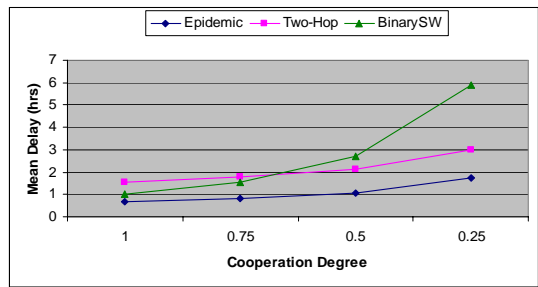

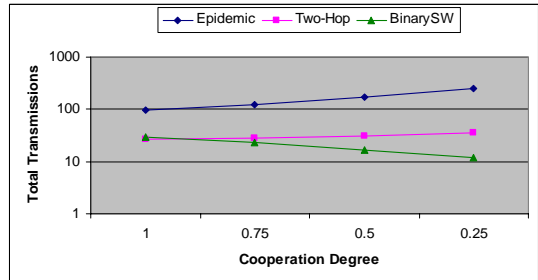Figure 1. Mean delivery delay as a function of Type I cooperation degree.

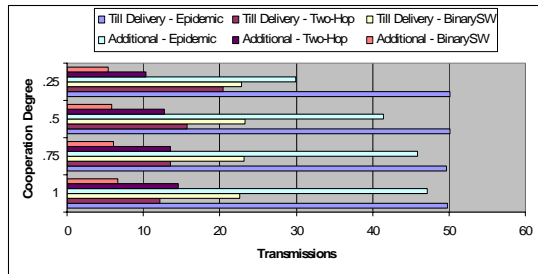Figure 2. Total transmissions (in log scale) as a function of Type I cooperation degree.

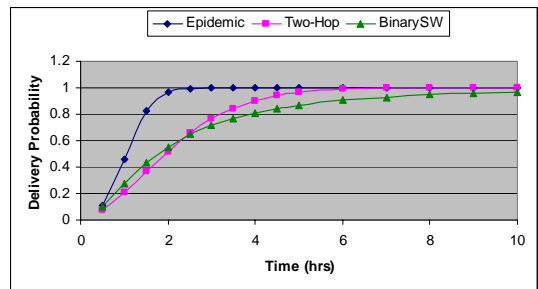Figure 3. Distribution of the overhead into its components.

Figure 4. Cumulative distribution function (*cdf*). (Type I cooperation degree of 0.5).

Clearly, epidemic routing always provides for the minimum delivery delay but at the expense of significantly more transmissions both under cooperative and non-cooperative environments. Concerning the other two algorithms, binary spray and wait achieves lower delivery delay in fully cooperative environments, but the rate at which the delivery delay of the two-hop relaying algorithm increases in non-cooperative environments is significantly lower in comparison with that of

spray and wait; as a result, there is a specific cooperation degree (approximately 0.75) below which the delay performance of the two-hop relaying algorithm outperforms that of the spray and wait. At the same time, the total overhead induced by the binary spray and wait algorithm decreases in less cooperative environments, contrary to the trend observed under the two-hop relaying algorithm. The above behavior may be attributed to the fact that in binary spray and wait a node might be responsible for spreading up to N/2 message copies; thus, the number of copies that are retained in the network may decrease rapidly under non-cooperative conditions.

Fig. 4 depicts the cumulative distribution function (*cdf*) of the delivery delay for the three algorithms, for N=K=100 and a cooperation degree equal to 0.5. It can be seen that although the two-hop algorithm induces a smaller mean delivery delay than the binary spray and wait, the latter outperforms slightly the two-hop relaying algorithm for time intervals less than approximately 3 hrs, in the sense that the probability of delivering the message within such a time interval is higher. This example reveals that the comparative performance of the two algorithms with respect to the mean delay or the probability of achieving message delivery within a delay bound may be different.

In the sequel, the sensitivity of the three algorithms to cooperation is investigated, using the two metrics as previously described. Figures 5 and 6 illustrate the first metric as a function of the cooperation degree; the metric has been expressed as the normalized mean delay (Fig. 5) and total transmissions (Fig. 6), with respect to the case of fully cooperative nodes.

As it may be seen, epidemic routing is the most sensitive algorithm concerning total transmissions while binary spray and wait is the most sensitive one as far as the mean delivery delay is concerned. On the other hand, the two-hop relaying algorithm is the least sensitive with respect to both the total transmissions and mean delay.

Figures 7 and 8 illustrate the second metric as a function of the cooperation degree (d); this metric is expressed as the normalized mean delay (Fig. 7) and total transmissions (Fig. 8), with respect to the network's FCE. As it may be seen, epidemic routing is the most sensitive algorithm concerning the total transmissions while binary spray and wait is the most sensitive one as far as the mean delay is concerned. On the other hand, binary spray and wait is the least sensitive with respect to the total transmissions and two-hop relaying algorithm is the least sensitive one regarding mean delay. Moreover, it can be concluded that for the two-hop relaying algorithm, the mean delay achieved in the original network is almost the same as in its FCE, while the total transmissions are approximately 1/d times the transmissions in its FCE. At the same time, for epidemic routing, the mean delay achieved in the original network is slightly lower than in its FCE while the total transmissions are less than $1/d^2$ times the transmissions in its FCE. In the case of binary spray and wait, the total transmissions remain almost the same while the mean delay is less than 1/d times that achieved in its FCE.
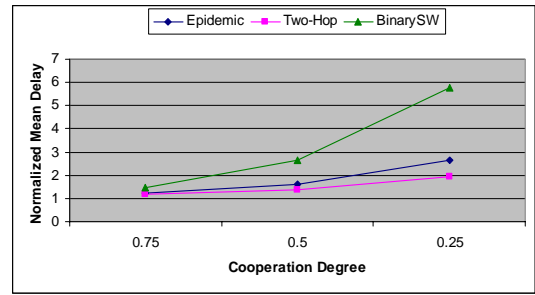

Figure 5. Normalized mean delay (wrt that in the case of a cooperation degree equal to 1) as a function of Type I cooperation degree.
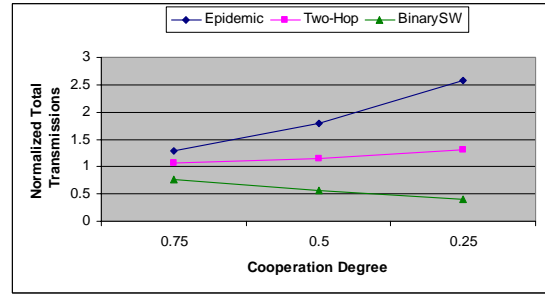

Figure 6. Normalized total number of transmissions (wrt that in the case of a cooperation degree equal to 1) as a function of Type I cooperation degree.
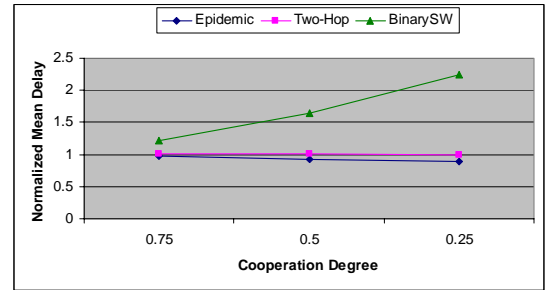

Figure 7. Normalized mean delay (wrt to its FCE) as a function of Type I cooperation degree.
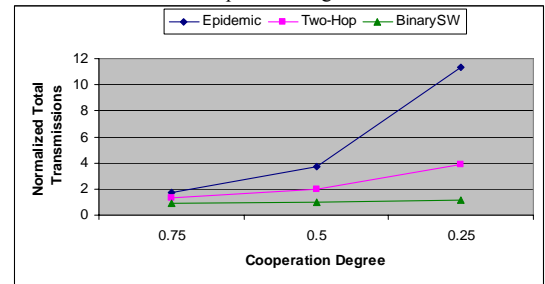

Figure 8. Normalized total number of transmissions (wrt to its FCE) as a function of Type I cooperation degree.

As it may be concluded from both sensitivity metrics, epidemic routing is the most sensitive algorithm regarding transmissions while binary spray and wait is the most sensitive one regarding delay. The two-hop algorithm is shown to be less sensitive with respect to both the induced transmissions and mean

delay, due to the fact that the always cooperative source node controls fully the copy spreading process and the lack of cooperation from a node has impact only on its own copy and not on the spreading of the other copies within the network.

B. Reaction to non-cooperative environments

In the previous section, we have studied the effect of cooperation on the performance of the routing algorithms under consideration. In this section, we assume that each node has knowledge of the degree of cooperation of the other nodes and investigate the effectiveness of a simple strategy that simply tries to avoid the nodes that are less cooperative than a pre-specified threshold.

More specifically, we assume that each node i drops the copy with probability $P_{drop,i}$ (Type I cooperation) or forwards it on behalf of some other node with probability $P_{forward,i}$ (Type II cooperation) and that this probability is known to the other nodes. We do not make any specific assumption about how the nodes acquire this information; this knowledge may become available through some reputation scheme (as those described in the literature mentioned in Section I) that works ideally and, thus, the exact value of $P_{drop,i}$, and/or $P_{forward,i}$, are determined. Alternatively, we may assume that the nodes inform their neighbors about their degree of cooperation; this would make sense, for example, in the case where the degree of cooperation of each node is proportional to its resources in a network of heterogeneous nodes.

Provided that a node is aware of the cooperation degree of each node it encounters, an issue that is raised is whether the node should avoid giving a copy of the message to nodes that will drop it with a high probability, for Type I cooperation (forward the message with a small probability, for Type II cooperation). By avoiding the most non-cooperative nodes, one would expect to save some transmissions at the cost of a small increase in the delivery delay.

We investigate this idea by setting a threshold based on which every node decides on forwarding a message copy or not; that is, the node will give a copy of its message to node i only if (1-$P_{drop,i}$) exceeds a threshold, for Type I cooperation (or if $P_{forward,\ i}$ exceeds a threshold, for Type II cooperation). For delay sensitive traffic, this threshold could be a function of the time elapsed since packet generation, since the more the elapsed time, the more it is worth to take the risk of giving the message to some node with a high P_{drop} (or small $P_{forward}$).

In Fig.9, the mean delivery delay of the three routing algorithms is illustrated as a function of the applied forwarding threshold in the case of Type I cooperation. The total number of transmissions is illustrated, for the three algorithms, in Fig. 10. The corresponding results concerning Type II cooperation are depicted in figures 11 and 12, respectively. All the above results refer to the case of a network of 100 nodes that have a degree of cooperation ranging from 0 to 1 (except for the source and destination nodes, which are assumed by default as cooperative)

such that the mean degree of cooperation throughout the network is 0.5. (In the previous section, P_{drop}=0.5 has been assumed for all the nodes, except for the source and destination nodes; this explains the small differences in some of the derived results although in both cases the mean degree of cooperation is equal to 0.5.)
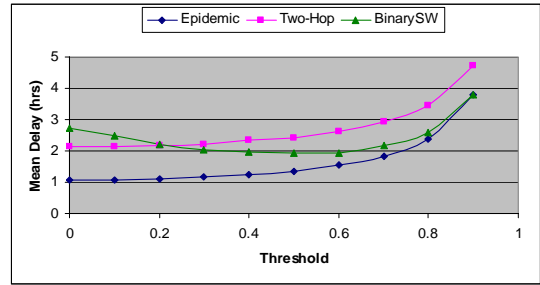

Figure 9. Mean delivery delay as a function of the applied forwarding threshold. (Type I cooperation with mean degree of .5)
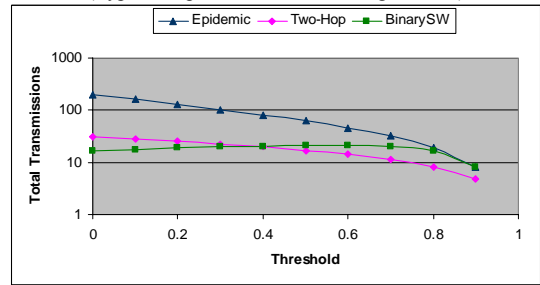

Figure 10.Total transmissions (in log scale) as a function of the applied forwarding threshold. (Type I cooperation with mean degree of .5)
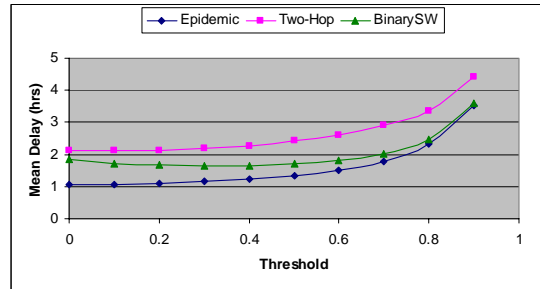

Figure 11. Mean delivery delay as a function of the applied forwarding threshold. (Type II cooperation with mean degree of .5)
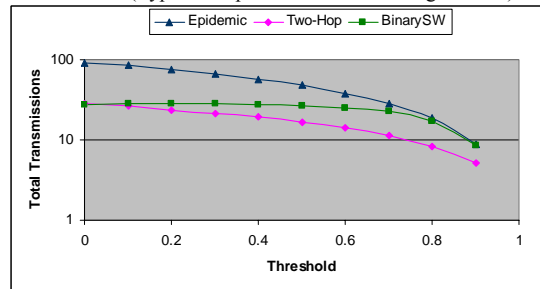

Figure 12. Total transmissions (in log scale) as a function of the applied forwarding threshold. (Type II cooperation with mean degree of .5)

When no threshold is applied, the two-hop relaying algorithm achieves a lower delivery delay than the binary spray and wait in case of Type I cooperation; for Type II cooperation, the behavior of each algorithm seems to resemble the one in fully cooperative environments. For both types of cooperation, the employment of the forwarding threshold leads to a significant reduction in the number of transmissions at the cost of only a small increase in the delivery delay, at least for a threshold less than the average degree of node cooperation. The only exception holds for the binary spray and wait algorithm in the case of Type I cooperation for which the inverse behavior is observed. More specifically, there seems to be a value of the threshold selected (~0.2 for the mean delay, and ~0.3 for total transmissions) below which the two-hop relaying algorithm achieves a lower delivery delay or/and induces more transmissions than the binary spray and wait and vice versa above that threshold. This may be justified by the fact that for relatively smaller values of the forwarding threshold, relatively more non-cooperative nodes will get and eventually drop a copy of the message; a node in binary spray and wait can be responsible for spreading up to N/2 message copies thus, the mean delay is seen to be worse than that achieved under the two-hop relaying algorithm, where the same non-cooperative nodes lose at most one copy. For this reason, and for the same values of the threshold, more transmissions take place under the two-hop relaying algorithm. For relatively higher values of the threshold, on the other hand, relatively more cooperative nodes are included in the spreading process; thus, the advantage of the spreading speed of the spray and wait algorithm (that clearly exists in a cooperative environment) starts to become evident.

Regarding Type II cooperation, the behavior of the algorithms resembles more the one in a fully cooperative environment. More specifically, the binary spray and wait algorithm always achieves a smaller delivery delay than the two-hop relaying algorithm but at the cost of a greater number of total transmissions. This may be justified by the fact that since message copies are not dropped, the binary spray and wait spreads the copies faster and, consequently, more copies are expected to have been spread by the time all copy carriers are notified of the message delivery.

IV. CONCLUSIONS

In this paper, the performance of epidemic, two-hop relaying and binary spray and wait routing is studied (both in terms of the achieved delivery delay and the induced transmissions) within the framework of a non-cooperative environment. Cooperation is modeled as the node's probability either to drop a message copy upon its reception (Type I cooperation) and/or to forward the message copy at a node encounter (Type II cooperation). The sensitivity of the nodes to cooperation is measured based on two metrics. It is shown that epidemic routing, which seems to outperform the others with respect to the achieved delivery delay at the cost of significantly increased transmissions, is the most sensitive one regarding the induced number of transmissions; the binary spray and wait algorithm is the most sensitive one regarding mean delay; on the other hand, the two-hop relaying algorithm is shown to be the least sensitive to cooperation

algorithm. Moreover, it is shown that by applying and fine-tuning a simple mechanism that takes advantage of the knowledge on the cooperation of the nodes within the network the performance of routing may be considerably improved.

REFERENCES

[1] S. Burleigh, K. Fall, V. Cerf, R. Durst, K. Scott, H. Weiss. L. Torgerson, and A. Hooke, Delay-tolerant networking: An approach to interplanetary internet, IEEE Communications Magazine, vol.41, no.11, Nov. 2003, pp.74-81.

[2] T. Spyropoulos, K. Psounis, and C. Raghavendra, Single-copy routing in intermittently connected networks, IEEE SECON, October 2004.

[3] J. Byers and Gabriel Nasser, Utility-based decision-making in wireless sensor networks, Tech. Report 2000- 014, 2000.

[4] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-2000-06, Duke University, April 2000.

[5] T. Spyropoulos, K. Psounis, and C. Raghavendra, Spray and wait: An efficient routing scheme for intermittently connected mobile networks, Proceedings of SIGCOMM 2005, August 2005.

[6] E. Altman, A. Kherani, P. Michiardi, and R. Molva, Non-cooperative forwarding in ad-hoc networks, Technical Report INRIA Report No. RR-5116, 2004.

[7] V. Srinivasan, P. Nuggehalli, C. Chiasserini, and R. Rao, Cooperation in wireless ad hoc networks, in Proceedings of IEEE Infocom, 2003.

[8] S. Buchegger and J. Le Boudec, Performance analysis of the CONFIDANT protocol: Cooperation of nodes - fairness in dynamic ad-hoc networks, in Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC), Lausanne, CH, June 2002.

[9] R. Molva P. Michiardi, Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks, Institut Eurecom Research Report, 2001.

[10] S. Zhong, Y. Yang, and J. Chen. Sprite, A simple, cheat-proof, credit-based system for mobile ad hoc networks, Technical Report Yale/DCS/TR1235, Department of Computer Science, Yale University, 2002.

[11] R. Groenevelt, Stochastic models in mobile ad hoc networks, Ph. D. thesis, University of Nice Sophia Antipolis, April 2005.

[12] M. Grossglauser and D. Tse, Mobility increases the capacity of adhoc wireless networks, IEEE/ACM Transactions on Networking, August 2002.

[13] C. Bettstetter, Mobility Modeling in Wireless Networks: Categorization, Smooth Movement, and Border Effects, SIGMOBILE Mob. Comput. Commun. Rev., vol. 5, no. 3, pp. 55–66, 2001.