Bayesian based selfish aware routing on Delay Tolerant Networks

Ricardo Oliveira*, António Duarte Costa†, Maria João Nicolau‡ and Joaquim Macedo§
Centro Algoritmi, Universidade do Minho, 4710-057 Braga, Portugal
pg17252@alunos.uminho.pt*, costa@di.uminho.pt†, joao@dsi.uminho.pt‡, macedo@di.uminho.pt§

***Abstract*—Delay Tolerant Networks (DTNs) aim to increase messages delivery ratio in environments where it is not possible to establish an end-to-end connection. Although the research of new DTN routing protocols has been gaining some relevance, those protocols usually assume that nodes in a network will collaborate.**

Nodes can behave selfishly, leading to the inappropriate use of resources, following up the malfunction of the network environment.

This paper presents an extension based on bayesian game theory to existing routing protocols. Each node tries to figure others node's type using the Naive Bayes classifier and behaves appropriately in order to achieve optimal results across the cooperative nodes. The regarded data through the exchangeable events between nodes can also be used to calculate each node's selfishness, assigning the acceptance and respective delivery probability of a message to its destination. The filter extension improved the delivery ratio of the cooperative nodes on selfish networks.

***Keywords*-DTN; Selfish Routing Protocols; Selfish Aware Routing; Bayes Classifier.**

I. Introduction

Over the past decade, the Internet's impact on the society has been increasing to the point of becoming an essential need for everyone's day-a-day. With the growing availability of mobile devices, the costs to maintain and install faster and broader centralized networks are also increasing [1]. On the other hand, rural areas, developing countries, military networks, or even in underwater or interplanetary networks lack the network infrastructure needed to offer continuous connectivity [2].

The connection between devices is made through the TCP/IP protocol which heavily relies in end-to-end and low delay connections. Those conditions are not always met. The lack of connectivity, commonly referred as disruption, may occur due to intermittent connectivity, long or variable propagation delays, low data rates and high error rates [3].

The high demand and the increasing cost of network structures led to increased interest on Delay Tolerant Networks (DTNs). These networks main goal is to offer data communication where it was not possible before, but it also improves communication on centralized networks by diminishing the infrastructures data load.

Originally called InterPlaNetary (IPN) Internet, DTNs aimed to improve the interplanetary communication; however, it was late discovered it is also adaptable to terrestrial networks.

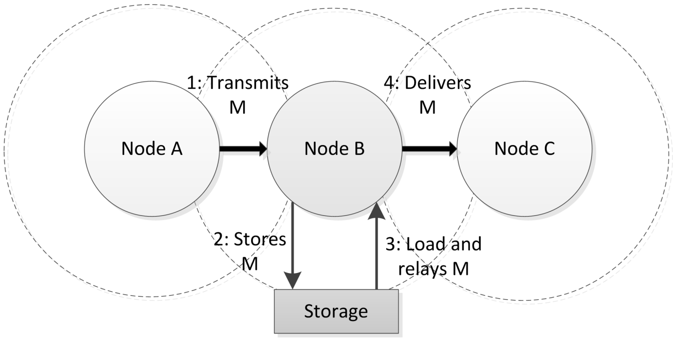

Fig. 1: Representation of the store-carry-forward technique on DTNs

As referred in multiple sources, [2], [3], DTNs consist in overlays, known as bundle layer, which operates above any of the communication protocols. Its mode of operation allows nodes with different underlying protocols and technologies to connect with each other. The bundle layer takes advantage of ad-hoc connections between several devices exchanging message between nodes with the store-carry-forward technique. In this method, messages are recursively stored and forwarded on intermediary nodes until eventually reach their destination as illustrated in Figure 1.

DTNs are a recent case of study and offer several opportunities for research such as more efficient routing protocols, security and fairness techniques, and data partitioning.

Several sources, [2], [3], [4], group routing protocols in two large categories: replica based protocols, and knowledge based protocols. The former ones make several copies of existing messages and forwards them to reachable nodes, i.e. flooding protocols. Those protocols are resource demanding which may lead to several problems. On the other hand, with the knowledge based protocols the movement of the networked nodes is predictable or even known, hence the exchanging of data only to the best known path. To perform the routing, those protocols require some knowledge of the network topology.

The majority of the routing protocols consider that every node in a network will behave as expected, but there are always deviations from users which will try to take advantage of the protocol and give priority on their messages. Those nodes can have a selfish or a malicious behavior, leading to the inappropriate use of energy, memory and network bandwidth, which gives an unfair advantage to the rest of the nodes [5],

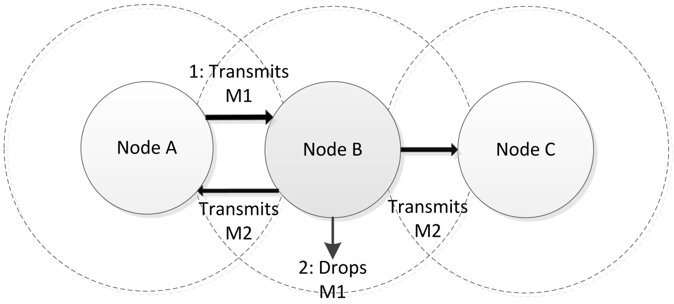

Fig. 2: Representation of selfish nodes on DTNs

[6], [7]. This incorrect execution of those nodes translates as a forced drop-page of unimportant received messages for them or the broadcast of too many messages to the rest of the nodes. Those nodes will not relay as expected and will drastically reduce the delivery ratio of the messages. This behavior is exemplified in the Figure 2.

This behavior can be controlled with incentive based routing protocols. In this paper, we propose a new extension based on bayesian games theory for existent routing protocols. No information is shared or considered between nodes other than the known interactions done between them.

This paper is structured as follows. Section II discusses the several available routing protocols and techniques to increase the delivery ratio on selfish network environments. Section III describes relevant considerations used on the design of the proposed routing protocol as well as the proposed algorithm. Section IV discusses results and compares routing protocols. Section V concludes the paper and discusses future work.

II. Related Work

Selfish nodes can be treated as a requirement, or as a deviation of the expected typical behavior of the routing protocol. There are several works which consider the presence of selfish nodes on DTNs [8], [9], [10], [11]. Nevertheless the following techniques and routing protocols are the most relevant for this work.

A. Coarse-Grained priority classes [5]

The goal of this technique is to minimize the effects of resource hogs originated by greedy nodes. The used approach only requires local information available within a DTN node.

The buffer management is structured around the concept of domains, prioritizing domain members to use its buffer space. A principal is a node or a set of nodes from the same domain. All messages coming from other nodes are classified as coming from a different domain.

If the buffer has enough free space it allocates the message, if not, it drops the message and then rejects them in order to free space. The technique was further expanded in order to diminish resources consumption of greedy nodes from the same domain. Three different methods were tested:

- The equal subdomains approach counts the number of distinct senders, whose messages currently occupy the buffer and then assigns a separate dynamic subdomain for each of them with a threshold.
- The usage-biased subdomains approach assigns a threshold for a given sender based on how many cumulative buffer space it used in the past.
- The penalty box approach identifies and blocks potential greedy node from the same domain until the buffer becomes uncongested.

It assumes cooperative/trusted nodes can be verified and authenticated by assigned authorities. Trusted nodes do not change their behavior.

B. Evolutionary forwarding games [7]

Defines the nodes forwarding policies according to the evolutionary game theory. Its main objective is to make a lower number of players with variable strategies not as successful as the rest of the nodes which maintain their strategy. The decision of the strategy to use with each node is made at the contact time. There are two modules responsible to calculate the utility of each one of the nodes: the activation control (AC), and the live time control (LTC).

- With the AC, during a local interaction between two nodes, each node can have two different strategies: to forward or not to forward its message to the destination. The utility of the nodes which have a forwarding strategy is calculated in conjunction with the estimation of energy cost of each message. Nodes with no forwarding strategy will not be considered.
- LTC prioritizes nodes which will keep the messages for the most of their time to live value.

C. RELICS [12]

This module assigns a rank to each one of the known nodes. It considers that every node behaves selfishly, trying to spend as less energy as possible but maintaining an acceptable delivery ratio. As a node relays more messages, its rank gets higher, and consequently, its messages priority increases on other nodes. The rank gets lower every time a node creates a new message. All the nodes involved in the relay of a message are rewarded.

The more messages a node wants to deliver, the more messages it needs to relay, consequently, the more resources it needs to spend. In order to solve this issue, each node is allowed to set a delivery ratio threshold which is used to activate its power saving features. A node's radio is enabled or disabled if its delivery ratio is higher than the threshold or not.

D. Discussion

The assumption of trusted information and trusting authorities is unrealistic under the consideration of heavily occupied and varying scenarios. We propose a new extension which only considers the node's observable information about the networks taxonomy and behaves accordingly with its made classifications.

III. Routing Protocol

The developed extension is based on game theory mixed with Naive Bayes classifiers. There is a two tier classification system. First, it classifies the scenario and then the nodes. In this section, we start by describing what kind of information can be considered as trustworthy, following the formulation of the bayesian game, the classifiers algorithm, and how buffers and connections are prioritized.

A. Trustworthy Information

Most knowledge based DTN routing protocols rely on shared information between nodes. On selfish environments, unless the information is given by a trustworthy node, that information should not be considered. With this proposed extension, we assume that the privacy and integrity of a message is assured by end-to-end encryption (by both the message's creator and its destination). Since neighboring nodes may only have access to the source address and destination, they can not modify the content of the sent messages; otherwise the destination node can not validate the incoming message and will discard it. This assumes nodes must share its public keys on a previously made contact with other ones to be able to exchange messages with them. With these considerations, it is then possible to share some statistically events based on the history of other nodes with both messages' creator and destination.

Despite these limitations, there is a considerable number of variables and events which can effect our knowledge of the network's cooperation. Considered outgoing events can be defined as:

- the number of contacts made by both Host and its neighbor
- the transmission of a Host's own message/ACK
- the retransmission of another node's message/ACK
- the direct delivery of a message/ACK from the Host to the neighbor
- the number of aborted transmissions
- the number of rejected receptions and their respective cause (the recipient was busy, the message is old, the message TTL (*Time To Live*) has expired, the recipient has low resources, denied based on a policy, and the recipient has no free space).

Similar data can be gathered regarding the incoming events but, the Host needs to consider every previously events to calculate how believable that data is.

A message and an ACK contain information about the node for whom the message creator sent the message to and the relay node who sent the message to the destination. The first is immediately before the transmission of the message. The second is in the ACK before being transmitted.

A fully cooperative node can provide information about all its known nodes. Due to the size that this information can reach, the number of requests made in an interval of time must be limited. The information provided by undefined, cooperative or fully cooperative nodes is always taken into consideration. Information given by fully cooperative nodes will be more valuable than the information given by cooperative or undefined nodes.

B. Bayesian games formulation

Cooperation can be stimulated, and disruptive nodes can be avoided with the application of the Game-Theory.

Game theoretic applications consider every interaction a player can make, being necessary to specify a set of rules for each one of the participants, as well as its outcome. There are a wide number of Game Theory types, and it is necessary to formulate them carefully for each case, but the common goal is to achieve fairness according to decision-makers actions. Those decision-makers, or players, are admittedly rational and noncooperative, that is, players perform actions which assure the best outcome for themselves. Those actions are essentially dependent of the available information from other players previous actions and its consequences. This set of information represents a player's strategy. The player strategies describe actions made at each stage of the game or associate probabilities to already known actions based on previous actions. [7], [13], [14]

Nodes are classified according to the player's gathered information. Non-cooperative nodes can give erratic information about their moves; hence the concept of Bayesian games is the most indicated for this work. Bayesian games considers the existence of imperfect information. In this case, the imperfect information is related to the node behavior. In order to know if a node is cooperative or noncooperative, we must assign it a type depending of the regarded information. This type is given probabilistically by an additional player called Nature (Ω). The rest of the problem is formulated as a normal game.

1) Strategies: In this Bayesian routing algorithm, rational nodes try to maximize their rank, and therefore, maximize the number of relayed messages across the network. In order to do this, the host will need to choose one of the four different strategies at every contact:

- $S1$: receive node N_i (Node i) message and send Host's message;
- $S2$: receive node N_i message and do not send Host's message;
- $S3$: do not receive N_i's message and send Host's message;
- $S4$: do not receive N_i's message and do not send Host's message.

2) Delivery probability and classification assignment: The Nature (Ω) of the Hosts game with N_i is given by the computation of the Cooperation Probability for N_i. In the very first contact with N_i, this value is 0.5 by default, hence the undefined classification. This value will increase and decrease as the Host, and N_i classification is changed along the game.

Given that the proposed protocol attempts to improve efficiency in routing messages on selfish networks, it is also necessary to take into consideration the quality of information provided by other network nodes and the information that each node can infer about the other. Thus, in this protocol a node only uses the network information provided by other nodes when they are considered fully cooperative. The delivery probability and the node classification of the remaining nodes are deduced with the exchange of messages. Some actions contribute positively, whereas others have a negative effect on both probabilities and classifications.

The information that contributes positively to the calculation of the delivery probability is deduced by:

- The number of messages relayed or delivered;
- The number of confirmation messages relayed or delivered.

The information that negatively contributes to calculate the delivery probability is deduced by:

- The number of aborted or denied messages;
- The number of messages sent.

However, the number of sent and rejected messages will only effect the nodes classification if they are higher than the threshold defined by the current scenario.

The bayesian game utility function is based on the following formulations:

$$PC\_new_i = PC\_old_i - (PC\_old_i) * P_iter \quad (1)$$

$$PC\_new_i = PC\_old_i + (1 - PC\_old_i) * P_iter \quad (2)$$

Where PC_new_i is the new cooperation predictability of node N_i and PC_old_i is node N_i old cooperation predictability. P_iter is a static value used to calculate new probabilities. The lower it is, the slower it converges into the correct type, but the more reliable it is. Ranges between 0.0 and 1.0.

C. Classifiers

As previously mentioned, the proposed algorithm uses two different Naive Bayes classifiers, more precisely, the Updateable Naive Bayes Classifier [15]. This specific version of the classifier enables the continuously change/increase of the classifiers training set, which makes the training set moldable with varying scenarios.

The host performs an action depending on the type associated to each one of the known nodes. When a new node is discovered, it is assigned the undefined type. This classification is maintained for a previously defined number of outgoing and incoming events (e.g. When the number of outgoing events or the number of incoming events is bigger than 30). Thereafter, a node's classification is recalculated at every new operation performed with a node.

A node can be classified as malicious, noncooperative, undefined, cooperative and fully cooperative:

- Malicious nodes are statically defined as identities who may drop, reject, or abort around 90% of the received messages, and broadcast all of it is messages.
- Noncooperative nodes are statically defined as identities who may drop, reject, or abort around 75% of the received messages.
- Undefined nodes are nodes which have around 50% of aborting messages. This could be caused by interference on the transmission of messages.
- Cooperative nodes accept around 75% of the sent messages and generally deliver messages to the Host.
- Fully Cooperative nodes accept most of the sent messages, generally deliver messages to the Host and have a low drop probability. Their classification is attributed only after a long set of positive classifications.

The classifiers take into consideration every type of event occurred in the exchange of data between nodes and consider the number of historic classifications.

Routing protocols where each message receives a delivery confirmation benefit for having a more assertive node's classification, but there is a bigger exchange of messages which decreases the buffer's capability.

D. Extended Buffer management

The messages stored in the buffer commonly have a FIFO order, which leads to the removal of the older messages when a new one is received. However, in this extension, the messages order is affected by both protocol ordering algorithm and the owner classification.

The Epidemic, Spray and Wait [16], and Prophet [17] routing protocols are common examples where messages are ordered in the same order as they are received. With our approach, messages are prioritized upwardly by a custom TTL assigned by the function:

$$Cooperation\ Index * Message's\ TTL. \quad (3)$$

E. Extended connection list

The list of outgoing messages are commonly ordered according to the specific routing protocol mechanics. In this extension, the sender's cooperation probability affects the message's order.

By default, messages are sent to every connection in the same order we have made a contact. This is the case of the Epidemic, and the Spray and Wait routing protocol. The list of connections is sorted with our extension, giving priority to the most cooperative routing protocols over the least.

Prophet sorts the connection list by delivery probability. Our sorting algorithm multiplies the delivery probability with the assigned Cooperation Index.

IV. Simulation and results

We have implemented and tested the extension on the well known Prophet routing protocol.

To test and evaluate this technique, we have carried several simulations with *The ONE* [18] simulator. *The ONE* is a complete, and commonly used framework to implement and evaluate DTN protocols. It is an open source Java solution, easily extensive, that supports mobility, event generation and message exchange. It includes DTN routing and application protocols and a basic notion of energy consumption. Visualization and analysis interfaces are also provided for importing and exporting mobility traces, events, and even entire messages.

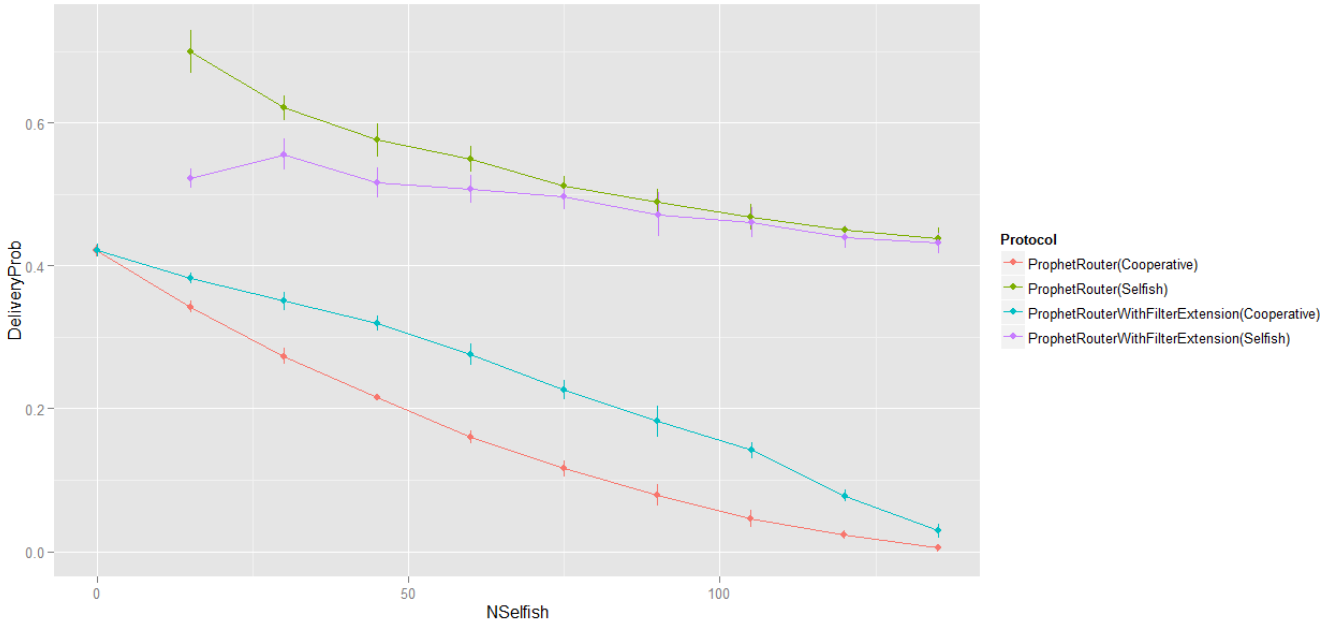

Fig. 3: Delivery probability in a 12 hours simulation

A. *Scenario description*

The proposed extension was evaluated in a scenario with similar specifications with the default one. It has a total number of 150 nodes (both cooperative and selfish). While selfish nodes only accept the reception of packets created by other selfish nodes, dismissing and dropping the reception and forwarding from cooperative nodes, cooperative nodes send and receive packets from any node.

The cooperative nodes use the proposed extension which classifies the nodes and decreases the priority in receiving and sending packets originally sent by selfish nodes. Messages from selfish also have a lower TTL.

Both type of nodes moved randomly through the default paths of the The ONE's default scenario.

Each simulation was executed with 30 different seeds. For each simulation i, there are $150 - 15i$ cooperative nodes and $15i$ selfish nodes. The first simulation started with 150 cooperative nodes and 0 selfish nodes, whereas the last one had 0 cooperative nodes and 150 selfish.

Each scenary simulated the interaction of nodes for 6, 12, and 18 hours.

B. *Evaluation*

The delivery probability refers to the probability of a message being delivered from a certain kind of nodes. Although selfish nodes always maintain a higher probability of delivery in comparison to the cooperative nodes, using the strategy proposed, the benefit of the selfish nodes is not as evident.

However, it is apparent that the probability of delivering by cooperative nodes (with and without strategy) decreases with the increasing of the quantity of selfish nodes in the network. This is because the number of nodes ready to relay cooperative messages becomes lower.

As can be seen, when the number of selfish nodes is 0, the probability of message delivery from this type of nodes is approximately 40%, while when the number of selfish nodes is about half of the total number of existing nodes in the network, the probability delivery is also about 20%.

Since selfish nodes only receive packets from other selfish nodes, as the number of selfish nodes increase, the number of delivery messages tends to resemble the number of messages delivered by the cooperative nodes when the number of selfish nodes is 0. These results are expected because selfish nodes treat selfish said messages in the same way that we deal with cooperative messages from other nodes.

Node type	6 hours	12 hours	18 hours
SE	0.4	0.5	0.53
S	0.43	0.52	0.55
CE	0.18	0.22	0.25
C	0.11	0.11	0.11

TABLE I: Delivery probabilities with and without extension filter

Table I represents the delivery probabilities of 75 Selfish, and 75 Cooperative nodes with and without the filter extension. SE and S stand for Selfish with Extension nodes and Selfish nodes, whereas CE and C stand for Cooperative with Extension nodes and Cooperative nodes.

As it can be easily noticed, Cooperative nodes in presence of Selfish nodes maintain a delivery probability of 11% as the time goes one, because they don't have any way to distinguish or filter the good from the bad behaving nodes, whereas Cooperative nodes with the Filter Extension begin to distinguish the type nodes, and start to prioritize their messages accordingly.

V. Discussion

Cooperative nodes which used the filter extension improved their delivery ratio, but the selfish nodes maintained their dominance. We expect to test the proposed filter on other routing protocols and to increase the aggressiveness of the extension so that selfish node's delivery ratio starts decreasing.

Furthermore, for constantly varying scenarios we plan to propose a new classifier which classifies a scenario according to previously similar history of events with similar delivery, contact and proximity ratios. At the beginning of the simulation, the set of scenario classifiers is empty, but, every n minutes, the current scenario information is recorded as well as the nodes classifications, making a new node's training set associated with the newly created scenario training set.

After the creation of a training set, when the next scenario starts, it uses the previously used classifier for 5 minutes (static defined variable), and then it finds the best match of the scenario it's associated node's training set. The finding would be made based on the information saved in the scenario's training phase.

Appendix A
Notation

Follows the notation and the meaning of the symbols used through the paper.

A. Players

- $Host$: The node which is choosing the strategy.
- N_i: The i node contacted in the network.

B. Bayesian Game symbols

- Ω: Nature of the game.
- $Npayoff_ij$: N_is payoff for doing j strategy.
- $Hpayoff_j$: Hosts payoff for doing j strategy.
- PC_def_i: N_is default cooperation predictability.
- PC_new_i: N_is new cooperation predictability.
- PC_old_i: N_is old cooperation predictability.
- P_iter: Static value used to calculate new probabilities. The lower it is, the slower it converges into the correct type, but the more reliable it is. Ranges between 0.0 and 1.0.

Acknowledgment

This work is partially funded by FEDER Funds through the Programa Operacional Fatores de Competitividade COMPETE and by National Funds through the FCT - Fundao para a Ciłncia e a Tecnologia (Portuguese Foundation for Science and Technology) within project FCOMP-01-0124-FEDER-022674

References

[1] G. Trecarichi, V. Rizzi, L. Vaccari, M. Marchese, and P. Besana, "Open-knowledge at work: exploring centralized and decentralized information gathering in emergency contexts," *Crisis*, no. January, 2009.

[2] V. Venkataraman, H. B. Acharya, H. Shah, and S. Lam, "Delay tolerant networking - a tutorial," *SciencesNew York*, 2009.

[3] F. Warthman, V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, H. Weiss, D.-t. N. Architecture, and et al., "Delay-tolerant networks (dtns)," *Networks*, no. March, 2003.

[4] S. Ali, J. Qadir, and A. Baig, "Routing protocols in delay tolerant networks a survey," *Knowledge Creation Diffusion Utilization*, pp. 70–75, 2010.

[5] J. Solis, N. Asokan, K. Kostiainen, P. Ginzboorg, and J. Ott, "Controlling resource hogs in mobile delay-tolerant networks," *Computer Communications*, vol. 33, no. 1, pp. 2–10, 2010.

[6] Q. Li, S. Zhu, and G. Cao, "Routing in socially selfish delay tolerant networks," *2010 Proceedings IEEE INFOCOM*, pp. 1–9, 2010.

[7] R. El-azouzi, F. D. Pellegrini, and V. Kamble, *Evolutionary forwarding games in Delay Tolerant Networks*. IEEE, 2010, pp. 76–84.

[8] A. J. Mashhadi, S. B. Mokhtar, and L. Capra, "Habit: Leveraging human mobility and social network for efficient content dissemination in delay tolerant networks," *Building*, pp. 1–6, 2009.

[9] L. Yin, H.-m. Lu, Y.-d. Cao, and J.-m. Gao, "Cooperation in delay tolerant networks," *Signal Processing Systems ICSPS 2010 2nd International Conference on*, vol. 1, p. V1202, 2010.

[10] U. Shevade and Y. Zhang, "Incentive-aware routing in dtns," *2008 IEEE International Conference on Network Protocols*, pp. 238–247, 2008.

[11] R. L. R. Lu, X. L. X. Lin, H. Z. H. Zhu, X. S. X. Shen, and B. Preiss, "Pi: A practical incentive protocol for delay tolerant networks," pp. 1483–1493, 2010.

[12] Y. S. Uddin, B. Godfrey, and T. Abdelzaher, "Relics : In-network realization of incentives to combat selfishness in dtns," *2010 18th IEEE International Conference on Network Protocols ICNP*, pp. 203–212, 2010.

[13] S. Keshav, "Mathematical foundations of computer networking," *Society*, pp. 177–202, 2005.

[14] S. Heap and Y. Varoufakis, *Game Theory: A Critical Introduction*. Routledge, 1995.

[15] G. H. John and P. Langley, "Estimating continuous distributions in bayesian classifiers," in *Eleventh Conference on Uncertainty in Artificial Intelligence*. San Mateo: Morgan Kaufmann, 1995, pp. 338–345.

[16] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, ser. WDTN '05. ACM, 2005, pp. 252–259.

[17] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7, no. 3, pp. 19–20, Jul. 2003.

[18] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE simulator for DTN protocol evaluation," *Proceedings of the Second International ICST Conference on Simulation Tools and Techniques*, 2009.