BIONETS

WP 4 – SECURITY

D4.3 Denial-of-Service Resistant Data Dissemination in BIONETS

Reference:	BIONETS/BUTE/wp4/2.0
Category:	Deliverable
Editor:	László Dóra(BUTE)
Authors:	Boldizsár Bencsáth, Levente Buttyán, László Dóra, Tamás Holczer, István Vajda (BUTE)
Verification:	Daniel Schreckling (HiTeC), Ioannis Koukoutsidis (NKUA)
Date:	March 28, 2008
Status:	Final
Availability:	Public

SUMMARY

In this Deliverable, we study data dissemination in BIONETS from a Denial-of-Service (DoS) point of view. More specifically, we consider data dissemination as a service provided collaboratively by the U-nodes and we focus on attacks that aim at denying this service. Our main objective is to identify the main sources of a potential DoS attack against the data dissemination process and to propose a method to prevent it.

In the first part of the Deliverable, we overview the data dissemination approaches introduced by other work packages in BIONETS. We show that a malicious attacker can degrade the quality of service, but due to the expected large size of a BIONETS network, a malicious attacker can have an effect only on a limited portion of the system, and thus, the effect on the quality of the services is also limited. In other words, due to the inherent redundancy in the system, an adversary can not completely prevent the spreading of messages. From another point of view, an adversary may inject fake messages into the system which slows down the dissemination of valid messages (i.e. generate unwanted traffic or spam), however some anti-spam techniques based content analyzing used in the Internet can be applied by the U-nodes to prevent the spreading of fake messages and to limit the effect of unwanted traffic.

A potential problem in BIONETS-like networks are that the quality of the service provided by the system heavily depends on the users' willingness to cooperate. Particularly, the users may act selfishly meaning that they download messages from other users that are interesting for them, but they deny storing and distributing messages for the benefit of other users. As potentially every user can be selfish, the effect of selfish behavior may have a much larger impact on the system than malicious attacks have. In particular, if the majority of the users behave selfishly, then the message delivery rate decreases considerably, and damages the quality of the service provided by the network.

Therefore, in this Deliverable, we focus on prevention of selfish behavior as the main source of DoS, and propose a novel data dissemination mechanism that discourages selfishness. Our proposed mechanism is based on the principles of barter. The users trade in messages, meaning that they can download a message from another user if they also provide a message in return. We analyze our proposed solution using a game theoretic framework, and show that it indeed discourages selfishness. More precisely, the analysis shows that it is worth for users collecting, carrying and disseminating messages even if they are not interested in them, which has a positive effect on quality of data dissemination. In particular, the results show that, in realistic scenarios, the message delivery rate considerably increases if the U-nodes follow the Nash Equilibrium strategy in the barter mechanism compared to the data dissemination protocol when no encouraging mechanism is present.

Contents

DOCUMENT HISTORY

Version History

Version	Status	Date	Author(s)
0.1	Draft	10 December 2007	Levente Buttyán, BUTE
0.5	Draft	22 December 2007	László Dóra, BUTE
0.6	Draft	04 January 2008	István Vajda, BUTE
0.9	Draft	29 January 2008	László Dóra, BUTE
1.0	Final	31 January 2008	Levente Buttyán, BUTE
2.0	Final	14 February 2008	László Dóra, BUTE

Summary of Changes

Version	Author	Date	Synopsis of Change
2.0	L. Dóra	14 Febr 2008	Final version
1.0	L. Buttyán	31 Jan 2008	Introduction and conclusion plus minor corrections
0.9	L. Dóra	29 Jan 2008	Denial-of-Service analysis of proposed protocols in BIONETS
0.6	I. Vajda	04 Jan 2008	Converge of the goodput
0.5	L. Dóra	22 Dec 2008	Barter-based cooperation
0.1	L. Buttyán	10 Dec 2007	Skeleton of deliverable

1 Introduction

The BIONETS network is a special type of opportunistic network where the transfer of messages from their source to their destination is performed by the intermediate nodes in a *store-carry-and-forward* manner. This means that the intermediate nodes carry the messages and pass them on to other intermediate nodes when they have a connection (e.g., when they are in vicinity).

In this Deliverable, the data dissemination process is viewed as a service, and the Denial-of-Service (DoS) attacks are directed against the data dissemination. More precisely, an attacker tries to stop or at least to slow down the message dissemination. This can be directed against a concrete message or for all the messages. The main objective of this Deliverable is to identify the most vulnerable points of the data dissemination process in the DoS point of view and to propose a method to prevent it. For this reason, first, we overview the data dissemination processes introduced by other research groups in BIONETS, and than we propose our mechanism which suits to the requirements addressed in the first part of the Deliverable.

There are three types of participant who may cause DoS attack in the system:

- An *external adversary* has limited access to the network, but it wants to degrade the level of data dissemination on purpose.
- A *malicious node* has all the rights that a participant of the network has, but it wants to degrade the quality of data dissemination.
- A *selfish node* is an internal node who assists in the data dissemination only if it can increase its utility.

The rest of the Deliverable is organized as follows. In Section 2, we overview the data dissemination protocols introduced in the project and we also analyze them from the Denial-of-Service resistance point of view. This section is divided into three subsections. First, we consider the erasure coding based solution for reliable data dissemination in Subsection 2.1. This is a general solution, it can rely on any data dissemination protocols. Than, we consider the data dissemination protocols categorized into two groups: pairwise connection based and broadcast based solutions, in Sections 2.2 and 2.3, respectively. The problem of the spam is considered in Section 2.4. After that, in Section 3 we present our proposed data dissemination mechanism which encourages the U-nodes to store, carry, and forward messages even if they are not interested in them. Finally, we conclude this Deliverable in Section 4.

2 Analysis of the BIONETS data dissemination mechanisms with respect to DoS

2.1 Data dissemination using erasure codes

Erasure codes were introduced in the context of BIONETS first in Deliverable 1.2.1 [1], later in Deliverable 2.2.1 [2], and finally a proposed protocol based on erasure codes is described in Deliverable 1.2.2 [3]. Erasure codes are also called (n, k)-codes, where n denotes the number of generated chunks after encoding a data (block) and k denotes the number of required chunks to decode the original data. The most important

property of the erasure codes is that the receiver is able to decode the original data from any $k \leq n$ of the chunks.

The benefit of using an erasure code in data forwarding is 1) to strength the reliability of data dissemination [1, 2] in the presence of selfish nodes and 2) to decrease the number of broadcasted packets in an environment where the participants of the communication vary in time, as detailed in [3]. The latter is case is out of scope of this Deliverable as it suggests an optimized solution for message delivery from the T-nodes to U-nodes, but not for the whole data dissemination process.

In the further case, the authors envisioned that a source node disseminates n chunks of a data. The property that k chunks are enough for any U-node to restore the original data helps to disseminate the data with high reliability even if selfish U-nodes reject to forward some chunks. Here, the erasure is caused by the selfishness of some U-nodes.

The authors only considered selfish but not malicious nodes. Selfish nodes simply do not forward the chunks of a valid message, but a malicious node may inject fake chunks into the network which may have negative impact:

- If a U-node obtains less than k valid and some invalid chunks, he/she will not be able to recover the original data. Moreover, if there is no redundancy in the original data, the U-node will be not able to discover the fault. Digital signature using the private key of the source on each chunk can provide protection against this attack, but it would make the encoding procedure even longer.

- If a malicious node injects into the network $k-1$ chunks of random data, the honest and altruistic U-nodes disseminate the $k-1$ chunks among them without being able to decode the data ever, because the analysis of $k-1$ chunks does not reveal any information about the validity of the original data, even if the chunks themselves are digitally signed.

2.2 Data dissemination based on pairwise connection

In BIONETS, three point-to-point based data dissemination protocols have been analyzed by looking at how robust the protocols are when selfish nodes are present in the network with different degree. The three point-to-point based data dissemination protocols are the epidemic routing [4], two-hop relay [5], and binary spray-and-wait algorithm [6].

Using epidemic routing, when two U-nodes encounter each other, they exchange the messages stored in their memory. This algorithm provides low transmission delay, but it wastes the memory and utilizes high bandwidth. Furthermore, usually the nodes are not able to exchange all the messages, since the connections are assumed to be short-term in BIONETS. The two-hop relay and the spray-and-wait algorithms correct this defect of epidemic routing mainly with limiting the number of message duplicates. A predefined system parameter (N) determines the largest number of message duplicates stored by the forwarding nodes. However, the distribution of the message duplicates among the forwarding nodes differs in the two algorithms:

- **Two-hop relay:** The source node passes a message to N forwarder nodes, but the forwarders are allowed to pass the message only to nodes that are directly interested in it.

- **Binary spray-and-wait:** A forwarder node passes half of its message copies to each encountered node that does not have that message. In case, only one copy remains at the forwarder node, the forwarder node passes that message only to nodes that are directly interested in the message.

In Deliverable 1.3.1 [7], the authors studied the impact of selfishness on the message transmission rate and delay. The selfishness means that the nodes reject to download (Type I) or to forward (Type II) the messages. In the model introduced in Deliverable 1.3.1, every node rejects to cooperate with some probability.

In Deliverable 1.3.2 [8], the authors assumed that the degree of the cooperation of each node is known by all other nodes. According to this, the authors proposed an algorithm to select the forwarder nodes to maximize the delivery rate and to minimize the latency.

The epidemic routing is the most reliable algorithm when selfish nodes are present. However, the main drawback of this algorithm is the high memory occupancy and the high bandwidth utilization compared to the other two algorithms. A malicious node can increase the number of unwanted traffic by behaving like a source of a message and setting the number of message duplicates to the maximum. In the case of two-hop relay, all the honest nodes become a forwarder of a specific message who encounters the malicious node, and in the case of binary spray-and-wait algorithm, all the honest nodes start to disseminate $N/2$ copies of the message. This kind of malicious behavior decreases the bandwidth, particularly in the case of the spray-and-wait algorithm where the malicious nodes can forward more copies of the messages to other nodes than it was defined in the mechanism description.

2.3 Data dissemination based on broadcast techniques

The other group of the data dissemination protocols introduced in BIONETS is based on broadcast techniques. Three of them are from the literature (Scalable Broadcast Algorithm (SBA, [9]), Generic Self-Pruning [10], HyperGossiping [11]), and two of them are developed in BIONETS (IOBIO [12, 8] , Multi Message SBA [8]). These protocols are analyzed in Deliverable 1.3.2 [8] with respect to the message delivery rate and the utilization of the bandwidth.

The main objective of the proposed protocols is to maximize the number of nodes reached with one broadcasted message, but also to minimize the unnecessary message and broadcast duplicates.

In SBA, the nodes are aware of their 2-hop neighbors. When a node receives a message, it starts a timer with a random value (within a predefined interval). Every time when the node hears the message, it registers who else received it. If the timer expires and someone did not receive the message in the neighborhood of the node, it rebroadcasts the message.

The Multi Message SBA (MMSBA) is extended with some features to be applicable in the BIONETS network. The nodes have to manage the changing neighborhoods and therefore they have to store the messages that they have already forwarded.

An adversary in the original SBA can easily prevent the dissemination of a message by broadcasting false information to its neighborhood. The steps of an adversary are the following: 1) The adversary collects the ID of two-hop neighbors. This information is sent in HELLO packets by the neighboring nodes according to the protocol. 2) Broadcasts a fake HELLO packet that claims that all the two-hop neighbors of

the adversary are its one-hop neighbors. 3) Later, when a message is broadcasted, the adversary immediately rebroadcasts it and the neighbors of the adversary register that all their neighbors received the message as the adversary claimed in the fake HELLO packet. Therefore, they will not rebroadcast the message again, at all. To be efficient, the adversary has to be near to the source of the message dissemination, otherwise the message will reach its destination through alternative routes.

When using the MMSBA, the nodes store the messages that they received even if all the known neighbor obtained them. They rebroadcast a message if they encounter a node which has not received the message according to the node's knowledge. In this case, an adversary is less effective.

To measure the effectiveness of the adversary — using SBA or MMSBA — placed at different points, some further investigations are needed.

In the generic-prune dissemination protocol, the nodes are aware of the subnetwork containing their k-hop neighbors. The messages are flooded among the nodes, but there is a self-prune mechanism, which is responsible for minimizing the message duplicates. A node i does not broadcast a message if there is a route between any node pair (u, v) within the k-hop subnetwork where all the nodes in the route have higher priority than the node i. The priority is not determined in advance. Its value can be chosen by a protocol designer. Moreover, the priority is not required to be in connection with any property of the message dissemination. E.g. the priority in some simulations was the number of neighbors. In this example, an adversary can claim higher number of neighbors to get higher priority. Note, that an adversary with higher radio range can get higher priority even if certificates from the neighbors are required. Furthermore, the adversary increases the priority of the neighboring nodes. Therefore, the k-hop neighbors will expect to receive the messages through the route where the adversary is placed. After that, if the adversary rejects to forward the message, it may have high impact on the message dissemination.

If the priority is based on the ID, an adversary can choose a large ID. However, if the neighbor of the adversary has a small ID, the other nodes will not prune themselves. Besides, if the priority is based on the reputation value, a node who rejects to forward messages will have low reputation value and low priority. Therefore, the choice of the priority value has a large impact on the security level of the message dissemination protocol. Further investigations are needed to choose the right priority value.

When nodes disseminate messages according to the Hypergossiping mechanism, they broadcast HELLO messages which include the stored messages. If a node detects a large difference between its memory and the memory reported in the HELLO message, then it rebroadcasts the messages are missing at the other node. With this mechanism, dissemination of messages are handled well when isolated islands join. However an adversary can easily persuade the other nodes to rebroadcast all the stored messages with broadcasted fake HELLO messages that claim that no message is stored in the adversary's memory. The honest nodes receiving the fake HELLO message rebroadcast all their messages exhausting their battery. Fortunately, the adversary has only local impact and it can not influence the nodes out of its radio range.

IOBIO implements a pull approach: first a node u broadcast a list of all or a part of the messages that it stores. If any node v in its neighborhood is interested in a message, the node v sends back a request message with the ID or IDs of the messages it is interested in. Node u waits until a timer expires and broadcasts the messages according to the requests.

Similarly to the Hypergossiping, in IOBIO an adversary has only local impact. Moreover, the adversary

is not able to choose which message will be broadcasted because of the pull mode. It can only force to make a message broadcasted even if no node is interested in it.

2.4 Spam in BIONETS

An adversary may inject fake messages into the system to slow down the dissemination of valid messages but with traditional anti-spam techniques based on only local decisions, similar to those used in the Internet, it is easy to prevent the spreading of fake messages. In the Internet these techniques are less effective than in BIONETS network, because they are applied in the end systems, and therefore they do not prevent the increased usage of the bandwidths. In contrast to this, in BIONETS networks, spam filtering can be implemented by the U-nodes which are not only end systems but forwarding nodes. Thus, unwanted traffic can be stopped immediately as it enters in the network, and it does not harm the entire system.

Considering some anti-spam techniques [13], the Bayesian distribution filters [14], statistical compression models [15], and using regular expressions are the techniques that suit to BIONETS networks. The common property of these techniques is that the U-nodes can decide locally if a message is spam or not only analyzing the content of the message.

- The Bayesian distribution filter algorithm binds a weight for each word in a message. The weights are adjusted by training the algorithm with spam and good messages. Later, the algorithm marks a message as a spam if its weight derived from the weights of the words of the message is heavier than a threshold.

- In case of statistical compression models, the analyzed messages are compressed with both spam and good-message models. If the message is compressed better with spam model than compressed with good-message model, the message is marked as spam.

- Widely used techniques are based on regular expressions. The messages are marked as spam if they match some predefined regular expressions. These expressions can be updated regularly in the Internet, but these can be updated also in BIONETS networks meeting other trusted U-nodes and exchanging the regular expression packages.

All these algorithms can be used together with rule-based ranking. In that case, each anti-spam algorithm gives a score to the messages. After summarizing the scores, the rule-based ranking decides if the massage should be mark as spam or not.

2.5 Conclusion

In this section, we have shown that a malicious attacker can degrade the quality of service. However, such an attacker has only local impact in a large scale mobile and distributed network. It is difficult for him to completely prevent the spreading of messages. From another point of view, an adversary may inject fake messages into the system to slow down the dissemination of valid messages but with traditional anti-spam techniques based on local decision, similar to those used in the Internet, it is easy to prevent the spreading of fake messages.

A potential problem in BIONETS-like network is that the quality of service provided by the system heavily depends on the users' willingness to cooperate. In particular, the users may act selfishly meaning that they download messages from other users that are interesting for them, but they deny storing and distributing messages for the benefit of other users. If the majority of the users behave selfishly, then the message delivery rate decreases considerably, and damages the quality of the service provided by the network. This has also been shown in [16] for the protocols described in Section 2.3. Therefore, a potentially dangerous source of Denial-of-Service is the proliferation of selfish behavior of the U-nodes. For this reason, we focus on this problem in the rest of this Deliverable. More specifically, in the next section, we analyze the problem and introduce a data dissemination mechanism that discourages selfishness.

3 Discouraging selfish behavior in data dissemination using barter

In the previous section, we showed that a malicious node is not able to have a large impact on the data dissemination. Meanwhile, the large number of selfish nodes considerably degrade the quality of data dissemination service. According to this, we propose a novel data dissemination mechanism resistant to selfish nodes. More precisely, our proposed mechanism is based on the principles of *barter*. The users trade in messages based on barter, and a user can download a message from another user if he/she can give a message in return. We expect that it is worth for the users collecting messages even if they are not interested in them. It is beneficial for the users to be able to give new messages to other parties in order not to skip any messages that the user is interested in. Thus, we expect that as a "side-effect", the messages disseminate better in the network.

This section of the Deliverable is organized as follows. In Subsection 3.1, we analyze the system without any incentives and determine the points where stimulating mechanism should be introduced. In the same subsection, we introduce the system model that is used to analyze the system with and without encouragement. We describe our barter based approach in Subsection 3.2, and we also extend the system model with the barter mechanism. For the analysis of the effects of selfish behavior in the system augmented with the barter mechanism, we introduce a game theoretic model in Subsection 3.3. In Subsection 3.4, we show and interpret the results of the barter game.

3.1 System analysis

In this subsection, we introduce the system model. Because of the complexity of the model, we use simulations instead of analytical tools. We show that there are scenarios where the message delivery has large latency because the U-nodes are selfish in a sense that they only store and forward messages that they are directly interested in. The aim of the analysis is twofold: 1) to prove that an incentive is required in the network to increase the message delivery rate and decrease the message delivery latency, and 2) to give a reference with which we can compare our subsequent solution.

3.1.1 System model

In our model, the users are placed in an arbitrary field. They own devices that have capabilities to communicate with other devices within their radio range. The used wireless technology can be Bluetooth, Wi-fi

or any wireless techniques that suits the BIONETS concept. The messages are generated and disseminated among the devices/users in the considered system, but each user is interested only in a small subset of the messages. The dissemination process is based on the store-carry-and-forward principle. A user and her device together is the *U-node*, and it is assumed that the message destination has no impact on the user's movement.

Each message has a type for each U-node. For simplicity, we distinguish only two types: primary messages and secondary messages. A message is a primary message for a given U-node, if the U-node is interested in the content of the message and secondary if the U-node is not. Note that a message may have different types for different U-nodes, as different U-nodes are interested in different contents.

These messages are generated by the *T-nodes*[1]. In our system model the time is slotted, and the T-nodes generate new messages with a fixed average rate, ϱ messages per time step. The T-nodes are static and each one stores only the most recently generated message, which can be downloaded at the cost of communication by any U-node that passes by the T-node.

A message has two main properties: the first one is the popularity attribute and the second one is the value of the message over time. The popularity attribute $0 < \zeta \leq 1$ describes the probability that a randomly taken U-node is interested in the message. We assume that T-nodes do not generate irrelevant messages, hence we consider $\zeta > 0$.

Each message has some value for each U-node. The value of a message is determined by its age. For simplicity, we assume that primary messages of the same age have the same value for the U-nodes. Without loss of generality, we assume that the value of a primary message at the time of its generation is one unit, and this is discounted in time, because messages lose their value over time. This is usually the case in the applications that BIONETS networks are envisioned for. The discounting function $\delta(t)$ describes the value of the messages over time. Obviously, it is difficult or impossible to find a discounting function which suits to each application. Therefore, we defined three different monotone discounting functions. These three functions are formulated in Equation 1, 2, and 3, and plotted in Figure 1. In the first case, the message value decreases linearly, in the second case, the messages evaluate exponentially, and in the last case, the messages loose their value suddenly, similarly to a step function.

$$\delta_0(t) = \begin{cases} 1 - \frac{t}{500} & \text{if } t < 500 \\ 0 & \text{else} \end{cases} \tag{1}$$

$$\delta_1(t) = 0.995^t \tag{2}$$

$$\delta_2(t) = 1 - \frac{1}{1 + 1000 \cdot (1 - \frac{1}{20})^t} \tag{3}$$

When two U-nodes get in the vicinity of each other, they interact in the following way:

1. The U-nodes exchange the list of the messages that they carry. The exchanged lists contain only the short descriptions of the messages (including their time of generation) rather than the messages themselves.

2. Each U-node u removes from the list $L_v^{(0)}$ received from v the messages that are not primary for node u, and the ones that u already stores in memory getting the list $L_v^{(1)}$.

[1]The results are not significantly different if we also allow the U-nodes to generate messages.

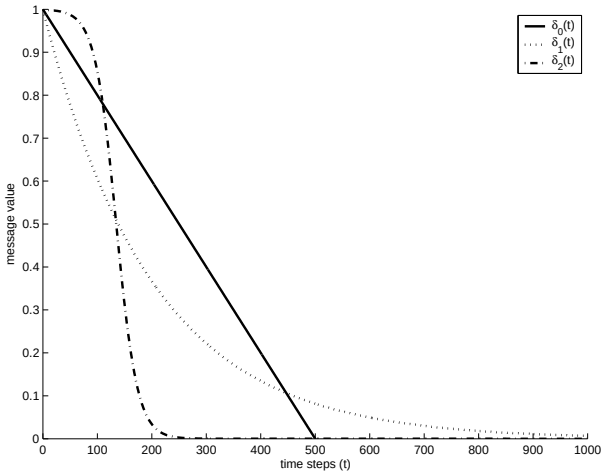


Figure 1: Primary message value

3. Each U-node u determines the value of the messages listed in $L_v^{(1)}$ based on their ages. Then, each U-node orders the messages contained in $L_v^{(1)}$ by their value in descending order. The resulting ordered list $L_v^{(2)}$ is the list of messages that u wishes to download from v.

4. U-node u and v download messages from the other party until they get new ones that they are interested in, or the connection is interrupted because the U-nodes move away from their radio range or they choose other parties to find new messages.

Connections can be interrupted because the U-nodes are moving and they leave the radio range of the other party. Therefore, in our model, the U-nodes are not able to exchange as many messages as they want but at maximum one message per time step. Hereby, we assume that a message exchange is completed in the time step or not started at all.

The assumption that only one message is exchanged per time slot implies that the U-nodes have to decide carefully which message they want to download in a time step. This can be viewed as an implicit cost of the system and we think that the effect of the implicit cost is much more important than the effect of different communication and storage costs. We think that the communication and storage cost is negligible compared to the value of the messages that a U-node could not download or could but with less information value. However, we take into account the memory constraints with the following mechanism: the U-nodes delete the messages from the memory whose value goes below the threshold D, $0 < D < 1$.

To measure the message delivery and delivery latency, we defined a formula for the goodput (see Formula 4 and 5). The notation is the following considering node i:

- m_i^t is the message that U-node i downloaded in time step t
- T_m is the time step when message m was generated
- δ is the discounting function described above
- $v_i(t)$ is the gain that U-node i gets in time step t

The goodput for U-node i is the sum of the gains in each time step normalized with the value that node i could obtain in the ideal case. In the ideal case, the U-nodes obtain all the messages at the moment of their generation. At the moment of the generation, the value of the messages is 1 as it is assumed above. Therefore, the maximum gain of U-node i is the number of generated primary messages ($M_i^P(t)$)

A U-node i gets $v_i(t)$ (see Formula 4) gain in time step t.

$$v_i(t) = \delta(t - T_{m_i^t}) \tag{4}$$

$$G_i(t) = \frac{\sum_{\tau=0}^{t} v_i(\tau)}{|M_i^P(t)|} \tag{5}$$

Note that the goodput is time and U-node specific. However, the goodput is statistically equal if all the U-nodes behave equally, but it can vary on choosing nodes behaving differently. Goodput also can vary over time, however we will show in Appendix A that the value of the goodput goes to a steady-state goodput. Therefore, we will consider the goodput of each U-node i in the steady-state conditions, denoted by G_i.

$$G_i = \lim_{t \to \infty} G_i(t) \tag{6}$$

Note that the maximum goodput is 1, which represents the case when the U-nodes receive all the messages that they are interested in at the time of their generation. This can be reached only by online networks which usually require installed infrastructure, operators and payment from the users. This contradicts the thought of the BIONETS.

3.1.2 Simulations

In our simulations, the fixed-number of U-nodes move in discrete time steps according to one of the two mobility models: the restricted random waypoint and SUMO (Simulation of Urban MObility, [17]) model.

In the restricted random waypoint model, the U-nodes move on field of size 20×20 unit. On the field, there are some special points chosen at random; these are called meeting points. Each U-node selects a meeting point randomly, and moves towards this meeting point with a fixed speed. When the meeting point is reached, the U-node stops and stays for randomly chosen time. Then, it chooses another meeting point and begins to move again. The nodes that happen to be at the same meeting point in the same time step are paired randomly and these pairs are able to download one message from each other in the above described way.

SUMO is an open source, realistic road traffic simulator. The vehicles start their movement from a randomly chosen place and they follow the traffic rules moving towards their destination also chosen at random in a predefined map. The vehicles move on the simplified map of Budapest (see Figure 2). The nodes can communicate with each other when they stop in the intersubsections similarly to the meeting points in the restricted random waypoint model.

Recall that in our system model, the messages are injected into the network by T-nodes that are static. In the restricted random waypoint model, the message nodes reside in the meeting points, whereas in SUMO the message nodes are placed in each intersubsection.

As we have already described the messages have ζ popularity value. When a T-node generates a new message in the simulation it determines which U-node is interested in it according to the popularity value. Thus, the message node sets the message to primary with probability ζ for each U-node.

We summarize the simulation parameters in Table 2.

Table 2: Parameter values for the simulations

Parameter	**RRW**	**SUMO**
Simulation length (time steps)	3000	
Number of mobile nodes	300	
Number of meeting/cross points	100	60
Number of message nodes	100	60
Message generation rate ϱ	0.01	0.0166
Simulation area	20×20 unit	see Fig. 2
Velocity (unit/timestep)	1	-
Probability of leaving a meeting point	0.1	-

While we defined the simulation parameters, we imagined the following scenario in the case of the restricted random waypoint model: There is a 3-storey building (mall) with 33 rooms (shops) in each storey and three persons (shoppers) on average and a message node (advertisement unit or seller) in each room, 300 mobile nodes and 100 message nodes in the building. In the case of SUMO mobility model, we also assumed to have 300 U-nodes, but in the map there is only 60 cross points, hence, 60 T-nodes are placed in every intersubsection. All the T-nodes generate one new message per time step on average. In the restricted random waypoint model the U-nodes stay at a meeting point for 10 time steps on average. We determined the length (number of time steps) of the simulation in an empirical way, and we take into account that the goodput have to reach the steady-state goodput.

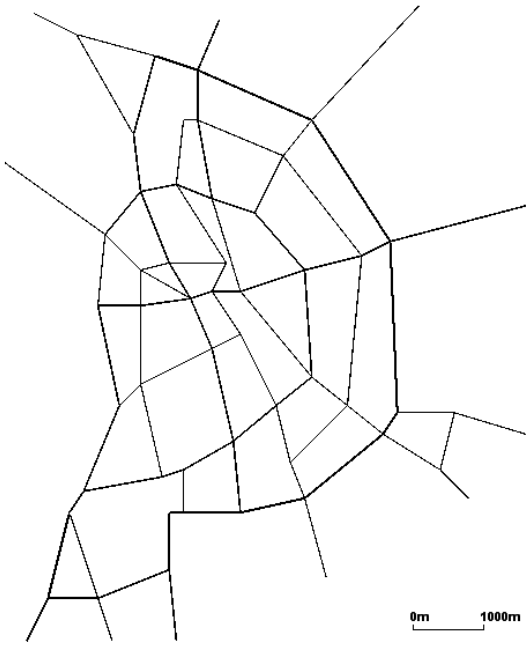
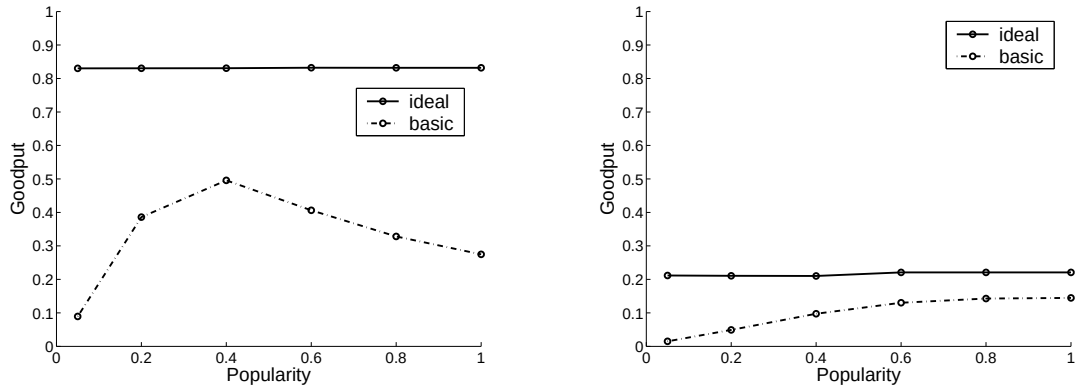


Figure 2: Simplified map of Budapest used in SUMO mobility model



(a) RRW (b) SUMO

Figure 3: Motivation results

We varied some of the parameters to study their effect on the results. As described above during simulation runs we used different functions for message devaluation. Besides this, for the sake of simplicity, we assumed that during a simulation the messages are generated with one predefined popularity attribute ζ, but more simulations were executed with different ζ values. Recall that $0 < \zeta \leq 1$. To reduce the complexity of our simulations, we use the following values of ζ: $\zeta = 0.05, 0.2, 0.4, 0.6, 0.8, 1$.

The main objective of these initial simulations is to prove that an incentive is required to increase the message delivery rate and to decrease the message delivery latency. Therefore, we run two kinds of simulations for every scenario: 1) one to get the goodput when the nodes behave selfishly and 2) another one to get the goodput in an ideal case. In the former case, the U-nodes strictly follow the protocol introduced in Subsection 3.1. This protocol corresponds to selfish behavior, because U-nodes download only those messages in which they are interested. In the ideal case, the U-nodes download all the new messages that they find in the memory of the connected node in one time step, both the primary and secondary ones. The latter case gives an upper bound on the achievable goodput of the U-nodes in the realistic case. Clearly, this upper bound is different from the theoretical maximum of 1, because the value of a messages decreases before reaching an interested U-node, if reaches it at all.

In both simulations, the U-nodes behave similarly in the same situation, hence they behave statistically equally in the whole simulation. Therefore, the goodput of the network is calculated getting the average of the goodput of each node.

3.1.3 Motivation

Results in the case of the restricted random waypoint model and SUMO can be seen in the Figures 3(a) and 3(b), respectively. In these figures, we show simulations where the discounting function is linear (δ_0), because the results show minor changes with other message devaluations.

In the figures, the goodput of the network is plotted against the popularity attribute value of the messages. To remind the reader, in the simulations in each parameter set, the messages have the same popularity value. The obtained average goodput is plotted against the popularity values which can be seen on the horizontal axis. The solid line shows the goodput of the network in the ideal case and the line with dashes and

dots shows the goodput of the network in the selfish case, when the U-nodes does not download secondary messages.

There are huge differences between the two mobility models. In the case of the restricted random waypoint model (shown in Figure 3(a)) the goodput is much higher than the one in the SUMO mobility model (shown in Figure 3(b)). This difference comes from the fact that the U-nodes choose one from the meeting points uniformly in the RRW case, however in the SUMO mobility model most of the U-nodes move through the largest intersubsections, whereas the smaller ones are bypassed or quickly crossed because of the smaller traffic. For this reason, the messages generated at smaller ones are passed to U-nodes with less probability, and these messages are deleted before spreading in the network. Recall that a message is deleted by all the carrying nodes if the value of the message goes below a D threshold.

When the U-nodes behave selfishly the popularity value has a large impact on the goodput. The more U-nodes are interested in a message, the more nodes download the message even if all the U-nodes are selfish. The more U-nodes download a message, the higher is the probability that a U-node will meet one who has downloaded the required message. We call this *selfish carrier effect* and it can be seen in the Figure 3(b), but not clearly in the Figure 3(a). There, the goodput increases with the increasing popularity until a specific value, but then the goodput decreases.

The reason for the decrease of the goodput when the popularity increases is the following: The goodput is a ratio as the Equation 5 shows. As one can see, the denominator (maximum value) can increase till infinity. While the nominator (obtained value) has an upperbound (even if it is difficult to determine in a concrete parameter set), because the nodes are able to exchange only one message in each time step.

To conclude the motivation subsection, we can state that the goodput is affected by two mainly independent, but opposite effects: the selfish carrier effect and the implicit cost. When the value of the popularity attribute is 1 the goodput is affected clearly by the implicit cost, whereas when the popularity value is near to 0 it is affected clearly by the selfish carriers. The implicit cost comes from a property of the system model, while the selfish carrier effect comes from the selfishness of the U-nodes. Therefore, we can state that an incentive is required to compensate the selfish carrier effect which mainly affects the goodput of the network when the popularity value of the generated messages is low.

3.2 Our barter based approach

Our approach to stimulate the cooperation of U-nodes is based on the principles of *barter*. More specifically, we require that when two nearby U-nodes establish a connection, they first send the description of the messages that they currently store to each other, and then they agree on which subset of the messages they want to download from each other. In order to ensure fairness, the selected subsets must have the same size, and the messages are exchanged in a message-by-message manner, in preference order. If any party cheats, the exchange can be disrupted, and the honest party does not suffer any major disadvantage (i.e., the number of messages downloaded by the honest party is at most one less than the number of messages downloaded by the misbehaving party).

Note that it is entirely up to the U-nodes to decide which messages they want to download from each other. They may behave selfishly by downloading only those messages that are of primary interest for them. However, selfish behavior may not be beneficial in the long run. In particular, the idea is that a message

that is not interesting for a U-node A may be interesting for another U-node B, and A may use it to obtain a message from B that is indeed interesting for A. In other words, the messages that are secondary for a U-node still represent a *barter value* for the U-node, and hence, it may be worth downloading and carrying them. Hereby, the messages can be viewed as an investment to get new primary messages later.

Recall that the U-nodes did not select the secondary messages from the list of the connected node when they selected the messages to download in the message exchange protocol introduced in Subsection 3.1.1. However, when the messages are exchanged according to the principles of barter, as it is mentioned above, it is worth downloading and carrying secondary messages also, even if the U-nodes are selfish (we will show that this statement holds). Therefore, the U-nodes need to compare the primary to the secondary messages when they order the list of the connected node.

Recall that there is no direct benefit of downloading a secondary message. It is worth to download to exchange later for primary ones. According to this, the value of the secondary messages is considered only when a node sorts the messages for download from another node. The value of a secondary message at the time of its generation depends on how the U-node values secondary messages with respect to primary messages, and it is discounted in the same way as primary messages. In other words, if for a U-node, secondary messages are worth SP units for some $0 \leq SP \leq 1$ at the time of their generation, then the value of a secondary message after t time units is $SP \cdot \delta(t)$. SP is called *secondary/primary ratio*. We have to emphasize that if $SP_u = 0$ than the U-node u does not download any secondary messages.

Note that in general, the value of a secondary message cannot be larger than the value of a primary message of the same age (i.e., $SP \leq 1$), because the primary message has the same barter value as the secondary message, and in addition, the U-node is interested in its content. However a specific secondary message which is more fresh than a specific primary message may have higher value and it can be exchanged to primary messages later, which will have higher gain all together.

We adapt the message exchange protocol according to the barter-based approach in the following way:

1. The U-nodes exchange the list of the messages that they carry.
2. Each U-node u removes from the list $L_v^{(0)}$ received from v the messages that u already stores in memory, and thereby obtains the list $L_v^{(1)}$.
3. Each U-node u determines the value of the messages listed in $L_v^{(1)}$ based on their types, their ages, and the secondary/primary ratio SP_u as described above. The list obtained in this way is denoted by $L_v^{(2)}$.
4. Each U-node u orders the messages contained in $L_v^{(2)}$ by their value in descending order. The resulting ordered list $L_v^{(3)}$ is the list of messages that u wishes to download from v.
5. The nodes exchange at most $\ell = \min(|L_u^{(2)}|, |L_v^{(2)}|)$ messages from the beginning of their lists on a message-by-message manner, where $|L|$ denotes the length of the list L. Thus, the number of exchanged messages is determined by the length of the shorter list or the duration of the connection.

To provide the circumstances for barter exchange, we assume that the U-nodes offer all their valid and only valid messages to download. On the one hand, it is not worth for any nodes to hide messages from other U-nodes, because it may decrease the number of messages that the U-node is allowed to download

from other U-nodes. On the other hand, we assume that a mechanism is presented in the system that prevents injecting fake messages with which a greedy U-node can increase the number of messages to download from other U-nodes. At the end of this section, we extend the barter mechanism with the usage of anti-spam techniques. With that mechanism, the node are discouraged to generate false messages.

The purpose of our analysis later in this Deliverable is to verify whether the barter based approach increases the goodput or not.

3.3 Game model

We introduce our proposed mechanism as a game to analyze the behavior of the U-nodes using game theory [18, 19, 20, 21]. Our objective is to prove that the network can reach high goodput using barter mechanism even if selfish U-nodes are present.

We define a non-cooperative game $G = [P, \{S_i\}, \{\pi_i\}]$. P is the set of the players, S_i denotes the strategy space of player $i \in P$, and π_i represents the payoff function of each player i. To be more precise π_i is the simplified denotation of $\pi_i(s_0, s_1, ..., s_{|P|})$, because the payoff of each player depends on the strategy played by the other players. This can also be denoted by $\pi_i(s_i, s_{-i})$ emphasizing the strategy of player i, where s_{-i} is the strategy profile of all the players except for player i.

In barter game, the players (P) are the U-nodes, and hence in the rest of this Deliverable, we will use the player and the U-node notation equally and alternately. The strategy space of each player is the secondary/primary ratio ($SP_i \in S = [0, 1]$), and each player $i \in P$ decides which $s_i = SP_i$ strategy she plays. The players choose their strategies in a way to maximize their goodput. Hence, the steady-state goodput is the payoff in the barter game.

$$\pi_i = G_i \tag{7}$$

In order to model the behavior of the selfish U-nodes, we introduce the concept of best response and Nash Equilibrium.

The *best response* of player i to the profile s_{-i} is a strategy such that:

$$B_i(s_{-i}) = \arg\max_{s_i \in S} \pi_i(s_i, s_{-i}) \tag{8}$$

If player i plays $B_i(s_{-i})$ strategy in game G it reaches the maximum from obtainable payoffs given that the other players play s_{-i}.

The pure-strategy profile s^* is a Nash Equilibrium if the following equation holds for s^*:

$$s_i^* = B_i(s_{-i}^*), \forall i \in P \tag{9}$$

Namely, in Nash Equilibria none of the players can increase their payoff changing their strategy unilaterally.

Symmetric game: $G = [P, \{S_i\}, \{\pi_i()\}]$ game is symmetric if each player has the same strategy space ($S_0 = S_1 = ... = S$) and their payoff functions are equal ($\pi_i(s_i, s_{-i}) = \pi_j(s_j, s_{-j})$ for $s_i = s_j$ and $s_{-i} = s_{-j}$, where $i, j \in P$). A symmetric G game can be denoted by $[P, S, \pi()]$ tuple.

As one can see, the barter game is a symmetric game, because the strategy space defined in the game is identical for all players. In our system model the nodes are not distinguished in a sense of their objectives

or with some special properties. Thus, they can maximize their payoff in the same way and they should get the same payoff in the same strategy profile.

In the analysis of the barter mechanism we are looking for the Nash Equilibria. We limited ourselves to find only pure strategy, symmetric Nash Equilibria because of two reasons. On the one hand, as the U-nodes are not aware of decisions of other U-nodes, they are not able to determine which strategy to choose without additional mechanisms as the strategies are different, and the nodes have to decide which strategy to choose to follow the Nash Equilibrium. On the other hand we assumed that each U-node is a player, which would lead to the analysis of a game with a $|P|$-dimensional strategy space, which is infeasible by means of simulations.

A symmetric game has a symmetric pure-strategy equilibrium according to paper [22] if the strategy space is a nonempty, convex and compact subset of some Euclidean space while the function of payoff is continuous in the strategy and quasiconcave. The strategy space is the interval $[0, 1]$, which corresponds to the conditions of existing symmetric pure-strategy equilibrium. Whereas, the properties of the payoff function are not verifiable, the results of the simulations will show that the conditions hold.

If we expand the Formula 8 and 9 according to the symmetric game and equilibrium, $\{s^*\}$ is Nash Equilibrium if the following equation holds for any player $i \in P$:

$$s_i^* = \arg\max_{s_i \in S} \pi(s_0^*, s_1^*, \dots, s_i, \dots), \text{where } s_u^* = s_v^* \forall u, v \in P/\{i\} \tag{10}$$

As one can see, it is easy to verify that a specific strategy profile $\{s'\}$ is Nash Equilibria or not. Considering any player $i \in P$ — without loss of generality $i = 0$, called *player null* — if it is worth to deviate this player, $\{s'\}$ is not the pure strategy, symmetric Nash Equilibrium, whereas if s' is the best response to player null than s' will be the best response strategy for all the other players also, as the players have equal payoff functions.

Therefore, to find the symmetric pure-strategy Nash Equilibria is not necessary to examine the whole $|P|$-dimensional strategy space, but 2-dimensional is enough. Thus, due to the symmetry of the game, the analysis is independent of the number of players.

3.4 Results

We run simulations to analyze the efficiency of the barter mechanism as we did in Subsection 3.1. The simulations were executed with the same parameters such that we can compare the barter based mechanism to the other two analyzed cases: 1) when the messages disseminate ideally and 2) when the nodes download only primary messages.

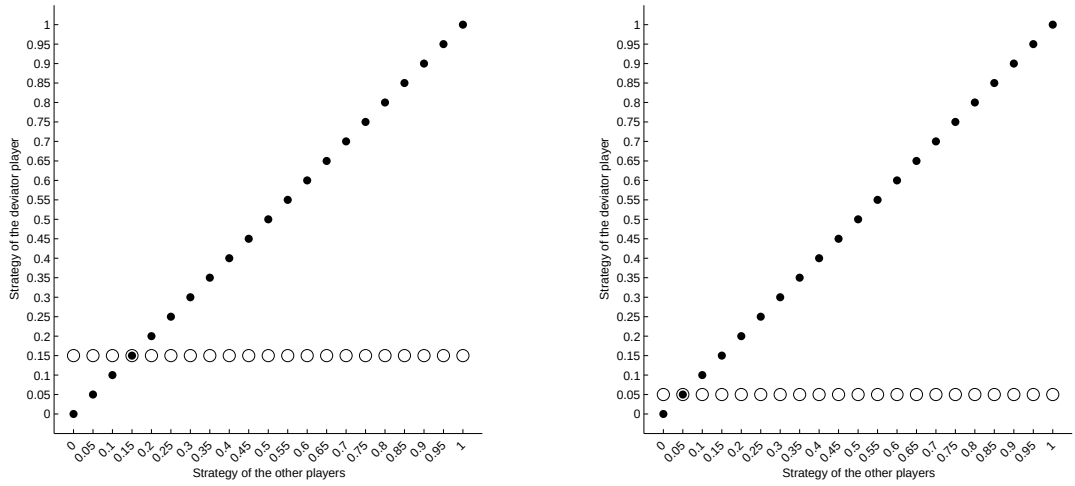
As we have already described, the U-nodes do not change their strategy during a game. Therefore, in each simulation run, the U-nodes play a predefined strategy chosen from discrete values of the strategy space. The discrete values are the values from 0 to 1 increasing by 0.05.

We run a simulation with a concrete parameter set six times, and we consider the average goodput of player null. The obtained goodput of the other U-nodes is irrelevant as it is described in Subsection 3.3.

Due to the above described discretization, each U-node's strategy can take 21 possible values. This means that we had to run $21^2 = 441$ simulations for each parameter setting in order to find the pure

strategy, symmetric Nash Equilibria. The best response function of some parameter settings can be seen in Figures 4(a) and 4(b).

In Figure 4, on the vertical axis, there are the strategies that player null can choose, while on the horizontal axis, the strategy space of the other players is placed. The Nash Equilibrium candidates are the strategy profiles where player null and the other players choose the same strategy; these are denoted by solid, black points in Figure 4. Whereas, the best response strategy of player null to a specific strategy profile of the other players is denoted by empty circles. E.g. in Figure 4(a) player null can get the highest payoff if its strategy is 0.15 mainly independently from other player's strategy. According to this, the Nash Equilibria is the strategy set where all the nodes play with strategy 0.15. The quasi deterministic best response appear in all parameter sets sometimes with different strategy, this is why we presented only two of them: one with the restricted random waypoint model and one with the SUMO, the messages devaluate according to the function δ_2 (see Formula 3) and the popularity of the generated messages is 0.2.



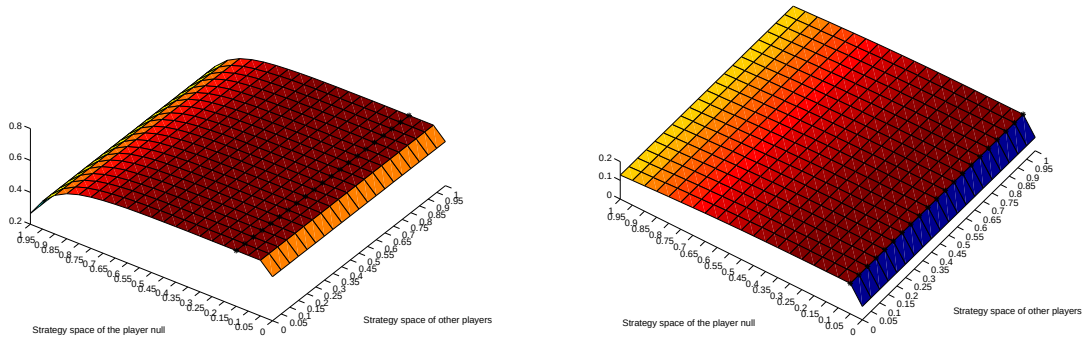
(a) RRW (b) SUMO

Figure 4: Best response

In Figures 5(a) and 5(b), the results of simulations are plotted in an extended form. In these figures the payoff of player null (vertical axis) is plotted against the chosen strategy of player null and other players (horizontal axes). The best response was calculated by these values in the following way: Player null always selects the strategy where the payoff is maximal besides fixed strategy of the other players. Recall that the other players choose always the same strategy. The best response is denoted by stars in Figure 5.

As one can see, the payoff of player null intensively falls down if player null does not cooperate ($s = 0$) or surprisingly, if it is too altruistic ($s = 1$). The nodes are encouraged to carry messages when the barter mechanism is used, because their goodput is higher if they do so (even if they are not directly interested in those messages). However, if the U-nodes are too altruistic and value the secondary messages as high as their primary messages, they help the other U-nodes (as it can seen also in Figures 5 with generally increasing payoff of player null when the other U-nodes are altruistic), but they suffer from goodput decrease.

To understand the reasons, we created some statistics during the simulations concerning the message exchange number and type. In Figure 6(a) we plotted the number of all message exchanges against the



(a) RRW (b) SUMO

Figure 5: Gain

strategies of player null and other players, and also classified the downloads by the type of the downloaded message (primary or secondary), these are plotted in Figure 6(b) and 6(c), respectively.

According to two message types (primary and secondary), there are three types of message exchanges between two U-nodes: 1) primary-primary, when two connected U-nodes download primary messages from the other, 2) primary-secondary, and 3) secondary-secondary; and there are two other types of message exchanges between U-nodes and T-nodes: 4) primary and 5) secondary message download. These are plotted in Figures 7(a), 7(b), 7(c), 7(d), and 7(e), respectively.

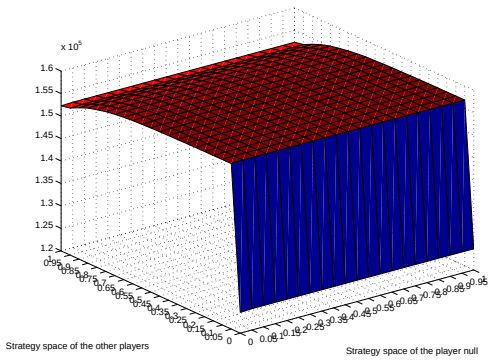
Note that there are message downloads, when the nodes download primary or secondary messages directly from T-nodes. The strategy of the players also effect the rate of message downloads from T-nodes as the more cooperative the U-nodes are, the more cases are when a U-node u meet another U-node v who forwards the message before the U-node u meets the T-node which have generated the message. We did not place the results in this document, because the effect on the direct download is marginal.

The Figure 6(a) shows that the message exchange deliberately decreases when the U-nodes do not cooperate at all. When the U-nodes do not cooperate, they also reach lower goodput, because the messages disseminate slower in the network than in the cases when the U-nodes cooperate ($s \neq 0$).

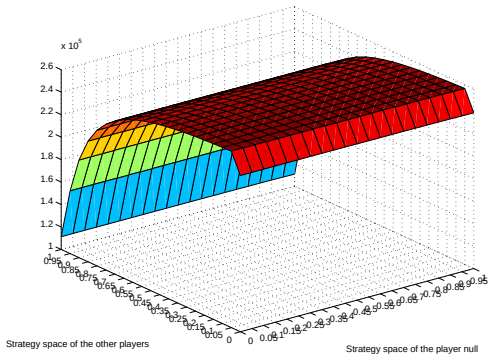
However, the U-nodes also reach lower goodput if they are too altruistic. The reason is the following: As one can see, when a player increases its value of strategy (the willingness of prefering some secondary to some primary messages), the number of obtained primary messages decreases while the number of obtained secondary message increases, whereas the number of message exchange does not vary appreciably (not taking into account when the U-nodes do not cooperate at all). This shows that the U-nodes following altruistic strategy do not utilize the investment of downloading secondary messages, but download more secondary ones.

To conclude the result of simulations, we can state that the strategy which is most beneficial individually – the Nash Equilibirium of the barter game – are s values that are near to 0, but not equal to 0. Therefore, it is beneficial to help the other nodes ($s \neq 0$) carrying their messages when the nodes exchange messages only in fair manner. However, if they are too altruistic, they download primary messages with less probability, and their goodput decreases.

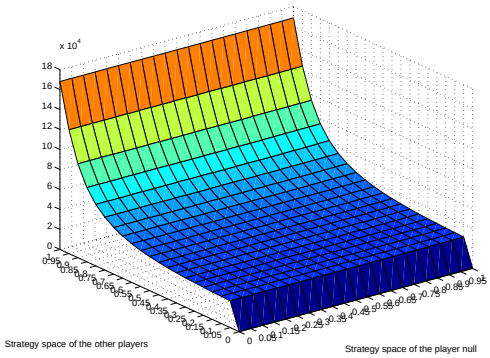
In Figures 9(a) and 9(b), the network goodput is plotted against the popularity attribute of the generated



(a) Exchange number



(b) Primary download



(c) Downloading number of secondary messages

Figure 6: Statistics

messages with restricted random waypoint and SUMO mobility model, respectively. It was done also in Figure 3, but these figures are supplemented with the network goodput in Nash Equilibrium of the barter game. As it can be seen, the barter mechanism eliminated the selfish carrier effect, however the implicit cost degrades the goodput in cases where the messages are generated with higher popularity attribute. The implicit cost is a system property, therefore it cannot be compensated.

The Figures 9(a) and 9(b) clearly show that the barter mechanism increases the network goodput in the networks where the popularity value of generated messages is low, while it does not decrease the goodput when the message popularity is high. Furthermore, when the popularity value is low the network goodput is as high as the optimal network goodput.

3.5 Supplement using anti-spam techniques

In Section 2.4, we mentioned some anti-spam techniques that suits to BIONETS networks. Here, we show how this techniques can be involved to encourage the dissemination of valid messages and to slow down the dissemination of spam messages.

Recall that the U-nodes before exchanging messages they determine an order of messages that they want to download. They take into account the age and the type of the messages. Basically, we defined two types: primary and secondary. The secondary messages usually have less value than the primary messages, they are rescaled with the secondary/primary ratio ($0 \leq SP \leq 1$) chosen by each U-node. However in

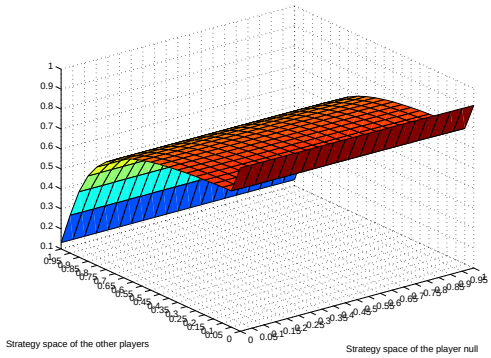
the ordering the spam marks can also be considered. If a U-node marked a message as a spam, the other party rescale the massage value with spam/primary ratio ($SPAM, 0 \leq SPAM \leq 1$). Typically, the spam/primary ratio is less than secondary/primary ratio ($SPAM \leq SP$) because the barter value of a spam is less than a valid message that a node is directly not, but the other nodes may are interested in. Note that the anti-spam algorithms sometimes mark messages even if they are valid. For this reason, it may be worth to download a message in spite of the fact that another U-node marked as a spam.

The U-nodes can be encouraged to use spam markers (which has low false negative and false positive rate) with interrupting the connection after downloading a message that seems to be a spam, but the other party have not marked. In that case, if a U-node uses an anti-spam algorithm which has a high false negative rate (marks the spam messages with low probability), the other parties will interrupt the connection frequently. Hence, the U-node will not able to download valid messages from the other parties. In contrast to this, if a U-node uses an anti-spam algorithm which has a high false positive rate (marks valid messages as a spam with high probability), it interrupts connections with high probability even it will not be able to obtain valid messages.

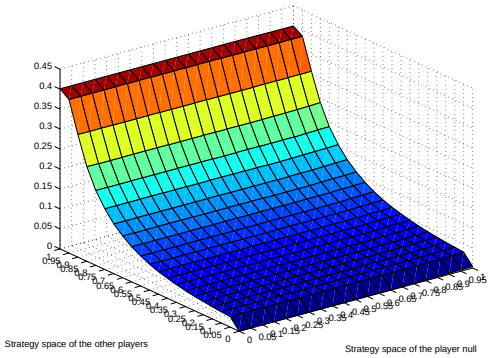
4 Conclusion

In this Deliverable, we studied data dissemination from a Denial-of-services point of view. In the first part of the Deliverable, we analyzed the data dissemination protocols introduced in the BIONETS project. We concluded that an adversary may have impact on data dissemination, however this impact is marginal compared to the potential impact of selfish nodes that deny to forward messages when they are not interested in them. The reasons are that the network is large and the message dissemination protocols are completely distributed. Therefore, to prevent the spreading of messages, the adversary must be physically present at many locations, which is almost infeasible. From an other point of view, an adversary may inject fake messages into the system to slow down the dissemination of valid messages but with anti-spam techniques based on the analysis of the message contents, used also in the Internet, it is easy to prevent the spreading of fake messages. We identified the potential selfishness of the U-nodes as the main source of Denial-of-Service, and hence we focused on discouraging selfishness in the data dissemination process.

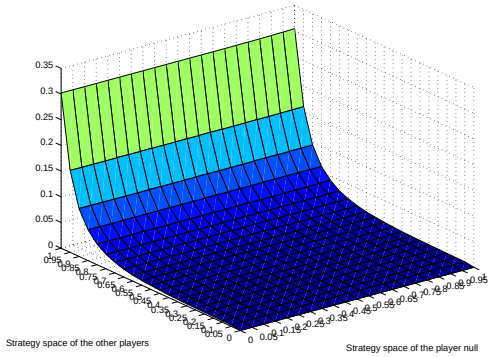
We proposed a new data dissemination approach, where the users trade in messages based on barter, and they can download a message from another user only if they also give a message in return. We analyzed out proposed solution using game theoretic techniques. We showed that it is worth for the users collecting and disseminating messages even if they are not interested in them, which means that our approach indeed discourages selfishness. The results show that, in practical scenarios, the message delivery rate considerably increases, if the U-nodes follow the Nash Equilibrium strategy in the proposed mechanism compared to the data dissemination protocol when no encouraging mechanism is present.



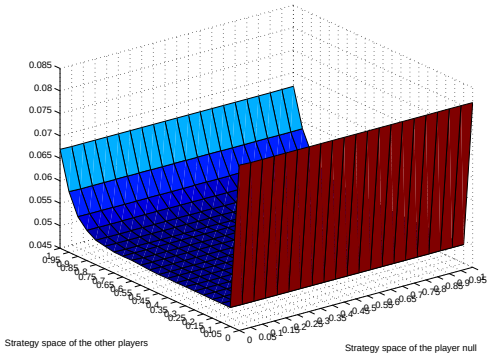
(a) Primary-primary exchange rate



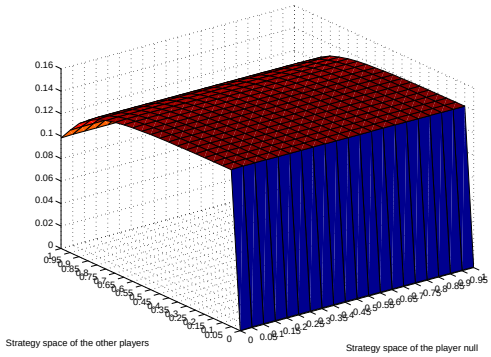
(b) Primary-secondary exchange rate



(c) Secondary-secondary exchange rate



(d) Primary downloaded from T-node rate



(e) Secondary downloaded from T-node rate

Figure 7: Message exchange statistics

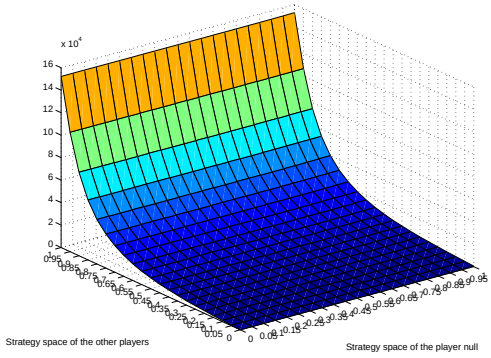
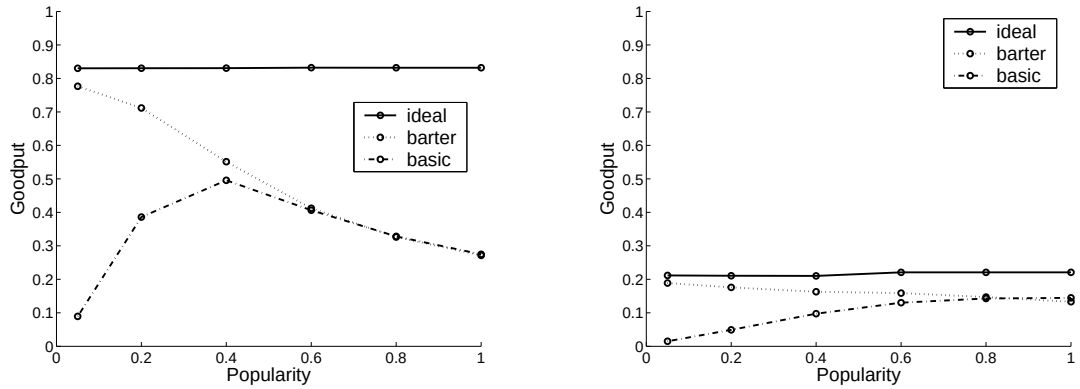


Figure 8: Number of preferring secondary messages to primary ones



(a) RRW (b) SUMO

Figure 9: Final results

References

[1] F. D. Pellegrini (Ed.), "Disappearing network infrastructure and design: Functionality and challenges," BIONETS (IST-2004-2.3.4 FP6-027748) Deliverable (D1.2.1), November 2006.

[2] P. Dini and A. Bassoli (Eds.), "Application of models of cooperation to network operation, design of p2p application, and social research through design," BIONETS (IST-2004-2.3.4 FP6-027748) Deliverable (D2.2.1), February 2007.

[3] D. Raz and F. de Pellegrini (Eds.), "Disappearing Network Autonomic Operation and Evolution," BIONETS (IST-2004-2.3.4 FP6-027748) Deliverable (D1.2.2), July 2007.

[4] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," 2000. [Online]. Available: citeseer.ist.psu.edu/vahdat00epidemic.html

[5] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad-hoc wireless networks," in *INFOCOM*, 2001, pp. 1360–1369. [Online]. Available: citeseer.ist.psu.edu/article/grossglauser01mobility.html

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

[7] S. Szabó (Ed.), "The initial mathematical models of new bionets network elements and algorithms," BIONETS (IST-2004-2.3.4 FP6-027748) Deliverable (D1.3.1), June 2007.

[8] ——, "Bionets simulation framework and initial performance analysis," BIONETS (IST-2004-2.3.4 FP6-027748) Deliverable (D1.3.2), August 2007.

[9] W. Peng and X.-C. Lu, "On the reduction of broadcast redundancy in mobile ad hoc networks," in *MobiHoc '00: Proceedings of the 1st ACM international symposium on Mobile ad hoc networking & computing*. Piscataway, NJ, USA: IEEE Press, 2000, pp. 129–130.

[10] J. Wu and F. Dai, "Broadcasting in ad hoc networks based on self-pruning," in *INFOCOM*, 2003.

[11] A. Khelil, P. J. Marrón, C. Becker, and K. Rothermel, "Hypergossiping: A generalized broadcast strategy for mobile ad hoc networks," *Ad Hoc Netw.*, vol. 5, no. 5, pp. 531–546, 2007.

[12] L. Bacsárdi, M. Bérces, E. Varga, T. Csvórics, V. Simon, and S. Szabó, "Strategies for reducing information dissemination overhead in disconnected networks," in *Proceedings of 16th IST Mobile and Wireless Communications Summit (MobileSummit)*, 2007.

[13] J. Goodman, G. V. Cormack, and D. Heckerman, "Spam and the ongoing battle for the inbox," *Commun. ACM*, vol. 50, no. 2, pp. 24–33, 2007.

[14] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A bayesian approach to filtering junk E-mail," in *Learning for Text Categorization: Papers from the 1998 Workshop*. Madison, Wisconsin: AAAI Technical Report WS-98-05, 1998. [Online]. Available: citeseer.ist.psu.edu/sahami98bayesian.html

[15] A. Bratko, B. Filipič, G. V. Cormack, T. R. Lynam, and B. Zupan, "Spam filtering using statistical data compression models," *J. Mach. Learn. Res.*, vol. 7, pp. 2673–2698, 2006.

[16] A. V. Antonis Panagakis and I. Stavrakakis, "On the effects of cooperation in DTNs," in *Proc. of The Second IEEE/Create-Net/ICST International Conference on COMmunication System softWAre and MiddlewaRE (COMSWARE)*, January 7-12 2007.

[17] "SUMO - Simulation of Urban MObility," http://sumo.sourceforge.net/.

[18] M. Félegyházi and J.-P. Hubaux, "Game theory in wireless networks: A tutorial," EPFL, Tech. Rep. LCA-REPORT-2006-002, Feb. 2006.

[19] D. Fudenberg and J. Tirole, *Game Theory*. MIT Press, 1991.

[20] R. Gibbons, *A Primer in Game Theory*. Prentice Hall, 1992.

[21] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*. Cambridge, MA: The MIT Press, 1994.

[22] S.-F. Cheng, D. M. Reeves, Y. Vorobeychik, and M. P. Wellman, "Notes on equilibria in symmetric games," in *In Proceedings of Workshop on Game Theory and Decision Theory*, 2004.

A Convergence of the goodput

In this section, we prove that the goodput of the nodes converges to a limiting value. This can also be seen in Figure 10 where the goodput of some randomly chosen U-nodes is plotted against the time. In Figure 11, the average goodput and its dispersion of all U-nodes is plotted against the time. After this analysis, we can state that the goodput obtained after a fixed number of time steps in simulation is the steady-state goodput.

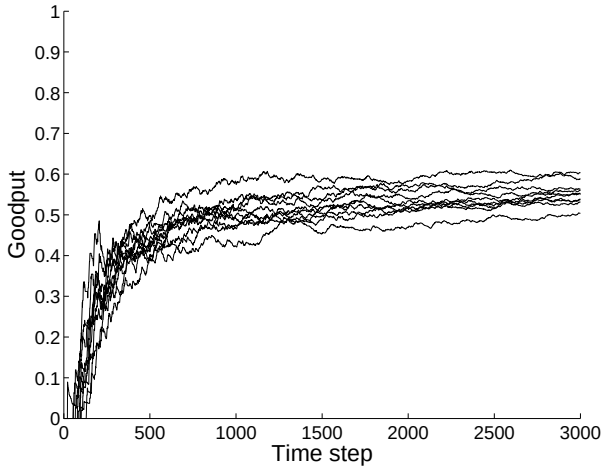


Figure 10: The converge of the goodput of some sample nodes

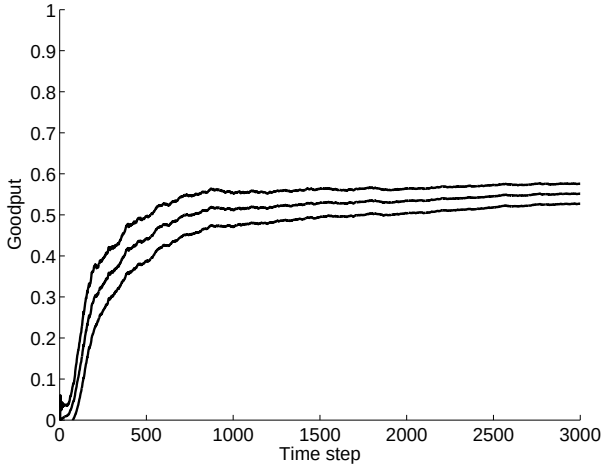


Figure 11: The converge of the average goodput and its dispersion

The state of the system described in Section 3 at time t is

$$\begin{aligned} s(t) = \{&B_1(t), B_2(t), \ldots, B_N(t), \\ &Z_1(t), Z_2(t), \ldots, Z_N(t), \\ &H_1(t), H_2(t), \ldots, H_N(t)\} \end{aligned} \tag{11}$$

where

- N is the number of nodes

- $B_i(t) = [m_{i_1}, m_{i_2}, \dots]$ is the buffer of node i, where the messages are stored.
- $Z_i(t) \in \{*, m\}$ is message stored in the memory of the T-node i, where $*$ denotes the case when no message is stored at time t, otherwise m stands for the generated message, which arrives from the — in principle — infinite space of messages.
- $H_i(t)$ is the position of node i on the field F.

We consider the finite state Markovian model in what follows.

Note, that the state space can be described by a deterministic mapping:

$$\begin{aligned} s(t+1) = F[&s(t), \\ &r_1(t+1), r_2(t+1), \dots, r_N(t+1), \\ &r'_1(t+1), r'_2(t+1), \dots, r'_n(t+1), \\ &r''_1(t+1), r''_2(t+1), \dots, r''_M(t+1)] \end{aligned} \tag{12}$$

where

- $r_i(t+1)$ is a random element used as an input by the algorithm to calculate the next step of node i ($1 \leq i \leq N$) on field F at time $t+1$
- $r'_i(t+1)$ is a random element used as an input of message generation of message node i ($1 \leq i \leq n$) at time $t+1$.
- $r''_i(t+1)$ is a random element used as an input of the node pairing in meeting point i ($1 \leq i \leq M$).

The random numbers are generated independently of the time.

Note, that the state transition mapping is time independent. The sequence of state random variables $S(0), S(1), \dots, S(t), \dots$ constitutes a discrete time homogenous Markovian chain. The transition matrix of the Markov process can be derived from Formulae 11 and 12.

As one can see the state space of the Markovian model described above is infinite, however with some feasible assumptions the model can be converted to a finite state model.

- Note that the memory of T-nodes was assumed to be unlimited in the whole Deliverable, however an upperbound can be defined. Recall that the U-nodes delete the messages if the message is older than T time step. Let the number of T-nodes be g. The greatest number of messages is generated if all the message nodes generate a new message in each time step. A message disappear from the system after T time steps. Therefore, the greatest number of messages that a node may store is $L = g \cdot T$. Hereby, $B_i(t) = [m_{i_1}, m_{i_2}, \dots, m_{i_L}]$.
- In the Markovian model described above, the m messages arrive from infinite space as there was no restriction for it. However, it is feasible to assume that the length of the digital contents that the nodes exchange is limited, let us assume to be l. In that case, the size of the message space is 2^l.

A Markov-chain is *ergodic*, if the following limiting value exist:

$$\lim_{n\to\infty} P_{ik}^{(n)} = P_k$$

these are independent of i and

$$\sum_{k=1}^{\infty} P_k = 1$$

As the classic theorem of Markov chains claims, a finite state homogenous Markovian chain is ergodic, if it is irreducible and aperiodic. Particularly, there is a time step t and a state j, such that state j can be reached from arbitrary initial state i with positive probability with time step t. The convergence to limiting distribution P_j is exponential, which means the following: let $P_{ij}^{(t)}$ denote the probability, that the Markovian chain starting from state i arrives at state j with t steps, furthermore let denote the stationary probability of state j, the difference $|P_{ij}^{(t)} - P_j|$ decreases exponentially when t tends to infinity (Theorem of Markov). In this case, uniform exponential bound exists for difference $|P_{ij}^{(t)} - P_j|$ independently of j.

In our model, the proof of the condition for ergodicity is the following: Assume the system is in an arbitrary state. We select a state k, let this state be the following, the buffer of the first node contains a single fresh message, while all other buffers are empty. Such a state can be produced the following way: First we empty all the buffers: the users move or stagnate at a fixed position such a way they escape meeting message sources. As the time passes the aging messages drop out from the buffer. Then the first node approaches a message source where it receives a message.

As it is shown above, our system is ergodic. Therefore, the convergence is exponential to the limiting distribution. The distribution of the stationary state is approached at exponential rate. As Equation 13 shows, the converge of the expected value derived from the state of the system — denoted by $\mathbb{E}f(S^{(t)})$ — is also exponential.

$$\mathbb{E}f(S^{(t)}) = \sum_{k=1} f(S_k)P_k(t) \xrightarrow[t\to\infty]{} \mathbb{E}f(S) = \sum_{k=1}^{|H|} f(S_k)P_k \tag{13}$$

The goodput of a node until time step t — as it is already described in Formula 5 — is:

$$G_i(t) = \frac{\sum_{t_j=0}^{t} \upsilon_i(t_j)}{\sum_{t_j=0}^{t} M_i^p(t_j)} \tag{14}$$

where the $\upsilon_i(t)$ is the gain that node i received in time step t, and $M_i^p(t)$ is the number of primary messages of node i generated in time step t.

As one can see, the goodput depends on the transient state of the system also, not just on the stationary state. In what follows we show that the effect of the transient state become negligible and fades away with exponential rate if the simulations run appropriately long.

$$G_i(t) = \frac{\frac{\sum_{t_j=0}^{t} \upsilon_i(t_j)}{t}}{\frac{\sum_{t_j=0}^{t} M_i^p(t_j)}{t}} \tag{15}$$

$$\frac{\sum_{t_j=0}^{t} M_i^p(t_j)}{t} \cong \zeta \cdot \varrho \cdot g \tag{16}$$

where ζ is the popularity value of the generated messages, ϱ is the rate of message generation, and g is the number of T-nodes. As one can see, in Equation 16, the denominator of goodput normalized by t is independent of time t.

Let the nominator of the goodput be the $\mathbb{E}f(S(t))$:

$$\frac{\sum_{t_j=0}^{t} \upsilon_i(t_j)}{t} = \frac{t'-1}{t}\left(\frac{1}{t'-1}\sum_{t_j=0}^{t'-1}\upsilon_i(t_j)\right) + \frac{t-t'}{t}\left(\frac{1}{t-t'}\sum_{t_j=t'}^{t}\upsilon_i(t_j)\right) \tag{17}$$

$$\frac{t'-1}{t} \xrightarrow[t\to\infty]{} 0 \qquad \text{and} \qquad \frac{t-t'}{t} \xrightarrow[t\to\infty]{} 1 \tag{18}$$

$$\mathbb{E}f(S) = \lim_{t\to\infty} \frac{1}{t-t'}\sum_{t_j=t'}^{t}\upsilon_i(t_j) \tag{19}$$

$$G_i = \lim_{t\to\infty} G_i(t) = \frac{\mathbb{E}f(S)}{\zeta\cdot\varrho\cdot g} \tag{20}$$

As we showed in this section, the system reaches its stationary state with exponential rate. Even though the goodput depends also on the transient state the effect of it is negligible after appropriately long simulation run as the effect of transient states fades away with exponential rate. By empirical observation, it is appropriate to consider the goodput after time step 3000 and the goodput will not change in the future considerably.