Revisiting Gossip-based Ad-Hoc Routing

Albana Gaba, Spyros Voulgaris and Maarten van Steen
Dept. of Computer Science, VU University Amsterdam
The Netherlands
Email:{agaba,spyros,steen}@cs.vu.nl

Konrad Iwanicki
Faculty of Mathematics, Informatics, and Mechanics
University of Warsaw, Poland
Email: iwanicki@mimuw.edu.pl

***Abstract*—We focus on a popular message dissemination protocol for wireless ad-hoc networks, GOSSIP3. Our contribution is twofold. First, we perform an extensive experimental evaluation of GOSSIP3 under fully utilized wireless channel and across diverse node densities. We identify the parameters of GOSSIP3 that need special configuration for the protocol to operate optimally. Second, we devise a self-configuration algorithm for GOSSIP3 that allows the protocol to work optimally for any network. We demonstrate through simulations that our protocol significantly outperforms the default configuration of GOSSIP3.**

I. INTRODUCTION

We are witnessing a quiet revolution as wireless devices become increasingly ubiquitous, silently intruding our daily lives. Their increasing pervasiveness can be attributed to further advances in miniaturization technologies, improved energy management, as well as better understanding of large-scale wireless networking and applications.

A number of applications can be envisioned, with tens of thousands of wireless devices embedded in surrounding physical objects. For instance, think of families or groups of friends carrying active bracelets when navigating in massive crowds (e.g., at an amusement park, a stadium, a public street event, etc.), that helps them stay loosely in close proximity. Think of smart tags attached to costly portable medical equipment in hospitals, or to parcels being (physically) routed through the complex hubs of a massive courrier company.

In this paper we consider large ad-hoc networks of wireless devices. A key observation is that, due to the unreliability of such networks and the lack of routing, *flooding* information to the whole network constitutes a rather common and fundamental operation. In other words, rather than transferring information from one node to another through a predefined route, techniques like flooding are often employed.

The major problem of flooding techniques, however, is a significant demand on network resources. As an illustration, consider the following scenario in which a family or group of friends attends the aforementioned amusement park of the future. To keep track of each other's presence, the bracelet of each group member regularly broadcasts presence information. To protect privacy — in particular, to prevent a maliciously-intended traffic analysis from tracking a lost or isolated child — the presence information is encrypted and flooded throughout the network, such that it can be recognized only by the members of the same group [1]. Allowing all nodes to insert new information at regular time intervals while successfully disseminating the information across the entire network requires efficiently accommodating the resulting high volumes of traffic within the available bandwidth budget.

In this paper, we revisit GOSSIP3, a well-known probabilistic dissemination protocol [2]. GOSSIP3 attempts to accommodate the heavy traffic resulting from gossiping by probabilistically forwarding data packets, effectively minimizing the *number of forwarding nodes*. At the same time, the protocol tries to compensate for packets that seem to be dying out by rebroadcasting them, thereby maximizing the *dissemination coverage*. Although the original GOSSIP3 can reportedly reduce the number of forwarders required for nearly perfect network coverage by 33% as compared to basic flooding [2], an improvement of such a magnitude requires fine-tuning the protocol's parameters. In a large ad-hoc network, however, this requirement may be too idealistic: there is neither a single parameter setting optimal for all networks nor even a setting optimal for all nodes in a given network.

Our contributions are twofold. First, we conduct an extensive study of the parameter set of GOSSIP3. More specifically, we investigate how different parameters affect the protocol's performance in terms of the number of forwarding nodes and dissemination coverage. We also analyze how different parameters interact with each other. Second, based on the study, we propose a novel self-configuration algorithm for GOSSIP3. In the algorithm, each node dynamically adjusts its local parameter settings using only local knowledge. We show through simulations that as a result of these autonomic local adjustments, the network as a whole achieves a nearly perfect dissemination coverage with a minimal number of forwarding nodes. In other words, our self-configuration algorithm removes the burden of manually configuring each node, thereby making GOSSIP3 an attrative all-to-all dissemination protocol for real-world ad-hoc wireless networks.

The rest of the paper is organized as follows. Section II briefly reviews the related work. In Section III we explore the parameter space of the GOSSIP3 protocol through simulations. Finally, in Section IV we introduce a way to self-configure GOSSIP3. Section V concludes the paper.

II. RELATED WORK

Broadcast protocols have been studied extensively in the past years due to their important role in many application scenarios. Plain message flooding protocols can incur a very high number of transmissions, large part of them being redundant.

On the other hand sophisticated protocols which reduce the number of transmissions dramatically by organizing nodes in a tree-like structure are impractical in dynamic scenarios. The key issue in broadcast protocols is the trade-off between the following contrasting goals: minimizing the message overhead and providing reliable message dissemination.

A range of broadcast protocols, known also as overlay-based, such as [3], [4], can be highly efficient in providing a very low message overhead by selecting the most connected nodes to act as forwarders. However, as shown in [5] these techniques can be sensitive to changes in network topology and packet collisions. Moreover, the creation and maintenance of the forwarding overlay often requires piggybacking the list of neighbors (e.g., in [3]), which is unrealistic for dense networks.

Haas and Nikolov [6] introduce a timing mechanism that allows covering all nodes in a network. This solution, however, is not suitable when multiple messages are being simultaneously disseminated from different sources, neither does it consider communication errors, such as MAC collisions or signal attenuation.

Simpler broadcast protocols, such as [7], [8], [9], [10], and [2], rely only on local information, such as feedback from neighbors (RSSI, location, message counter) or on probabilistic methods. This category is robust to topology changes, as it requires only local information. However, they are very sensitive to the setting of parameters. GOSSIP3 [2] is a well-known probabilistic, counter-based protocol, which is built as a result of an extensive study on gossiping properties in ad-hoc networks. It has been designed around a set of three parameters, which allow it to be simple and suitable for a variety of network configurations and densities. However, the original work on GOSSIP3 lacks a broad evaluation of the parameters in various densities and the effect of radio communication effect (MAC collisions and attenuation).

III. EXPLORING THE PARAMETER SPACE IN GOSSIP3

In this section we present the original GOSSIP3 protocol, and we explore the relation between its configuration parameters and network density. We start by describing our experimental setup.

A. Experimental Setup

We run simulations using the OMNeT++ Mobility Framework simulation environment. We consider random networks of 529 nodes with average distances between each other Δ: 5m, 10m, 15m, 20m. The traffic at the MAC layer is emulated by continuous transmissions of dummy packets at each node, whereas the MAC buffer delay is emulated by making sure that a new dummy packet is generated when the number of packets in the MAC buffer is smaller than five. Moreover, for each network density the MAC layer is set to transmit at a pace that has been prior selected to generate optimal traffic: maximal goodput. In order to have a more uniform distribution of the bandwidth and fairness among the nodes, we eliminate the border effect by using a 2D plane that is modeled as the surface of a torus.

Only one node, in addition to the artificially generated traffic at the MAC layer, inserts 200 data items to be propagated with a frequency of two data items per second.

The radio power is set to low (1mW), resulting in a maximum communication range of 50m. The following table shows the characteristics of each network configuration.

Avg. Node Distance	Avg. Neighbors	Network Diameter	MAC TX Success Ratio
$\Delta = 5m$	105	2	0.39
$\Delta = 10m$	19	5	0.52
$\Delta = 15m$	10	10	0.53
$\Delta = 20m$	5	27	0.53

B. The Gossip3 Protocol

GOSSIP3 constitutes an extension of probabilistic forwarding that aims to combine higher coverage of the network with reduced traffic volume. This is achieved by avoiding to forward every packet by every node.

More specifically, the operation of GOSSIP3 is determined by the following three configuration parameters:

- **Parameter p:** Upon reception of a packet, a node decides to forward it with a pre-configured probability, p. Consequently, with a small value of p, only a small fraction of nodes forward the packet.
- **Parameter m:** To compensate for packets that, due to probabilistic forwarding, may not reach all nodes, if a node has decided not to forward a packet, it waits for a short time interval snooping for the traffic on air. If within that interval the node does not hear at least m neighbors forwarding the packet, at the end of the interval, it will forward the packet despite its initial decision.
- **Parameter k:** Finally, to further minimize the chance of a packet dying out early, nodes within the first k hops from the source node that generated the original packet always forward the packet (i.e., for them $p = 1$).

In the original paper, $m = 1$ and $k = 1$ were claimed to be sufficient for most scenarios, while $p = 0.65$ was argued to provide the best performance. In the following section we revisit these claims by studying the impact of m, p, and k on coverage, traffic, and dissemination latency, for diverse network densities.

C. Impact of parameters m and p

We conducted an extensive series of experiments to explore the parameter space of GOSSIP3. More specifically, we tested all combinations for initial forwarding probability $p \in \{0\%, 10\%, 20\%, \ldots, 100\%\}$ and compensation parameter $m \in \{0, 1, 2, 3\}$. We tested each combination of p and m on four topologies of different densities, namely $\Delta \in \{5m, 10m, 15m, 20m\}$. In all these experiments, parameter k was fixed to the value 1, as suggested in the original GOSSIP3 paper.

The impact of p and m was tested with respect to three metrics: the coverage of the network, the induced traffic, and the dissemination latency.

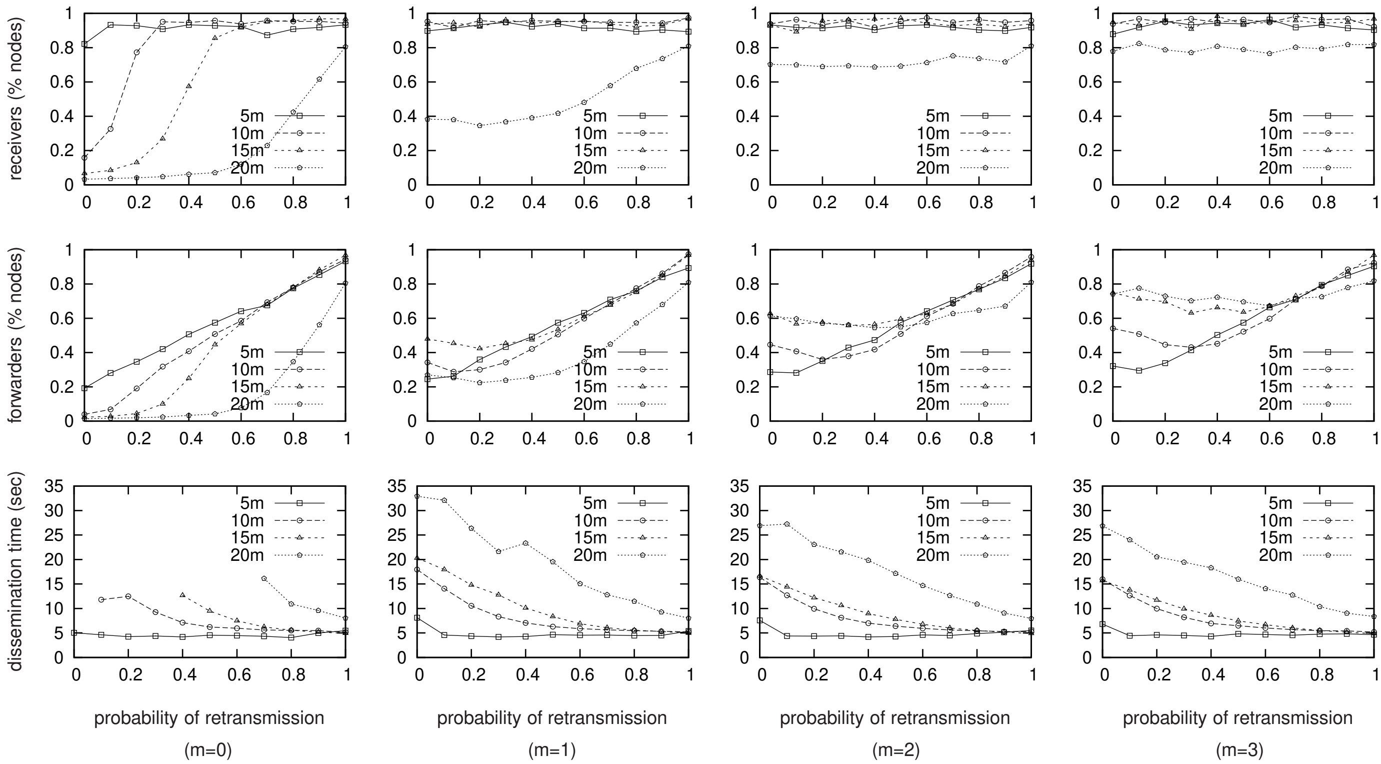


Fig. 1. Coverage, forwarders and latency for various m, p and various network configurations; $k = 1$

Impact on coverage: The top row of Fig. 1 illustrates the effect of parameters p and m on network coverage. When $m = 0$, we see that the forwarding probability, p, has a clear impact on the dissemination coverage. It is not hard to see why: when a node decides *not* to forward a received packet, this decision is final, as the compensation mechanism is disabled. Consequently, for low values of p, packets die out prematurely, effectively reducing the dissemination coverage. In denser networks this effect is limited, as a single node deciding to forward a packet is enough for a number of nodes around it to receive it. Topologies of lower density suffer significantly more.

In contrast, for $m \geq 1$ (still in the top row), the initial forwarding probability, p, has negligible effect on the dissemination coverage. This is a direct consequence of GOSSIP3's compensation mechanism, which compensates for insufficiently forwarded packets. The only exception is for the sparsest topology ($\Delta = 20m$) for $m = 1$. This is a special case, as the network is marginally connected, so a single node's decision *not* to forward a packet may result in a number of further nodes not receiving the packet at all.

In sparse networks we notice that in order to improve coverage, it is necessary to either start with a high probability, such as $p = 1$, which implies no compensation, or any probability (surprisingly even $p = 0$) in combination with compensation parameter $m = 3$.

Impact on traffic: The middle row of Fig. 1 shows the effect of parameters p and m on the amount of traffic induced by the dissemination of a single message.

As each node forwards a given message at most once, we measure traffic as the number of nodes that forwarded it, which essentially denotes the number of times a given message was transmitted. We express that number as a fraction over the whole number of nodes in the network.

Here we see that p strongly affects the number of forwarding nodes. Indeed, high values of p imply that nodes receiving a packet forward it with high probability, probably resulting in redundant transmissions, especially in dense topologies. On the other hand, for lower values of p, nodes receiving a packet are most likely *not* to forward it in the first place. As observed earlier, though, GOSSIP3's compensation mechanism forces certain nodes to deterministically forward insufficiently forwarded messages, resulting in no loss of coverage. Clearly, by means of the compensation mechanism, for low values of p nodes do not forward received messages unless "necessary".

Surprisingly, also setting the probability too low (e.g., as low as $p = 0$) may lead to higher traffic. When hardly any node decides to forward a packet, several nodes in the neighborhood jump in to compensate for it, leading to more transmissions than necessary.

The minimum number of forwarders is reached for $p = 0.1$, $p = 0.2$, or $p = 0.3$. Interestingly, due to the compensation phase, the same set of parameters $p = 0.1$ and $m = 1$ works best for most of the network densities, providing high coverage and a minimal number of forwarders. Although it appears as a win-win situation, the price to pay is higher latency, as we will see below.

Impact on latency: Finally, the bottom row of Fig. 1 depicts the effect of parameters p and m on the time needed for a message to reach all nodes.

Here we make two observations. First, low density topologies experience higher dissemination latencies. This is expected due to their higher diameter, which demands more propagation hops to reach out to the most distant nodes.

Second, the value of p has an inverse effect on the dissemination latency. This is expected too, as high values of p mean that nodes opt to forward messages instanly upon reception. For lower values of p, though, dissemination relies more on the compensation phase. Often, therefore, a node forwards a message only after the compensation timeout has expired, and it has observed fewer than m transmissions of that message so far.

D. Impact of parameter k

Parameter k dictates the number of (initial) hops for which nodes should forward a message deterministically, preventing a message from dying out before having reached a critical initial mass of nodes.

As mentioned earlier, in the original paper, Haas et al. claim that $k = 1$ works best for many networks. However, in our experiments (not plotted) we observe that performance improvement for $k = 1$ is insignificant. Moreover, in dense or very dense networks, $k = 1$ results in an explosion of message forwards by all first-hop neighbors of the source.

E. Discussion

Our experimental analysis in this section showed that GOSSIP3 requires a different set of parameters depending on the network density. In a practical setting, the network density may not be known a priori, or may be difficult to estimate. Our goal is to come up with a single mechanism that allows nodes to adaptively adjust their behavior to function optimally in diverse scenarios.

"Is it possible to get optimal performance on any network density without preconfiguration?". The following section addresses this question.

IV. SELF-CONFIGURATION OF GOSSIP3

We have shown that different network configurations exhibit different problems when it comes to information dissemination by GOSSIP3. In short, in dense networks, there are enough potential candidate nodes to forward data broadcast by a node, and these nodes are also well connected with each other. Therefore, high dissemination coverage comes virtually for free, and the key challenge is to avoid wasting bandwidth on redundant transmissions, which can be achieved by limiting the number of nodes that ultimately forward particular data. In contrast, in sparse networks, a node broadcasting data has few candidate nodes that can forward it further, the wireless links to those nodes are of poor quality, and the nodes are rarely connected with each other. Consequently, even if many nodes forward data, there are virtually no redundant transmissions, and instead, guaranteeing reasonable dissemination coverage becomes the main issue.

These conflicting requirements cannot be satisfied by a single parameter configuration for GOSSIP3. Worse yet, a single configuration may not be optimal even for a single network, especially if the density of the network is not homogeneous.

We address this problem by introducing algorithms in which each node self-configures its parameters depending on its current environment to maximize the performance of GOSSIP3. Our algorithm consists of two components — self-configured probabilistic forwarding and a self-configured compensation mechanism — which we discuss next.

A. Self-configured probabilistic forwarding

Our first idea is to let each node dynamically adjust its probability p of forwarding data based on the number of other candidate nodes that can forward the data if necessary. One of the benefits of this mechanism is that a network can dynamically self-configure after a deployment with a probability value that is most suitable for that particular deployment. Moreover, each node can choose a different, custom forwarding probability, which may be important especially in heterogeneous networks with sparser and denser regions. Finally, any mobile node can automatically reconfigure itself when moving between sparser and denser regions of the network.

1) Choosing forwarding probability: We have devised the self-configuration algorithm for p based on our empirical resuts. More specifically, using the aforementioned experimental data, we have identified those values of p for which GOSSIP3 achieves the best performance, that is, a maximal coverage with as few forwarders as possible. Assuming that the compensation mechanism is active ($m > 0$) and that initial nodes are not forced to always forward data ($k = 0$), the best performing values of p are as follows: for $\Delta < 10m$, $p = 0.1$, for $10m \leq \Delta \leq 15m$, $p = 0.2$, and for $\Delta > 15m$, $p = 0.4$. Whereas, the compensation parameter is $m = 1$ for $\Delta \leq 15m$ and $m = 3$ for $\Delta > 15m$. In Fig. 2, we plot the results from the experiments corresponding to these best performing configurations. Each point in the figure corresponds to a single node in a single experiment, with points belonging to the same experiment being represented with the same color and shape (dots, squares, crosses, etc.). Each point represents the final forwarding probability of a node as a function of the number of the node's neighbors (the local density of the network). The final forwarding probability is computed for a node as a ratio of the number of unique data items forwarded by the node to the number of unique data items received by the node during an entire experiment. The probability thus encompasses the forwarding probability p and the probability of compensating for data dying out. Oversimplifying things, the plotted final forwarding probability for a node may be thought of as the value of p the node needs to deliver an optimal dissemination performance without triggering the compensation mechanism.

Using the empirical values we have devised a heuristic for assigning p to a node depending on the local density of the network. More specifically, we have fed the empirical values to data analysis software[1] to obtain a best-fit function. The resulting function, representing a Weibull Model, is as follows:

$$p(N) = 1 - 0.87 \cdot e^{\frac{-50}{N^{2.3}}}$$

[1] CurveExpert Professional

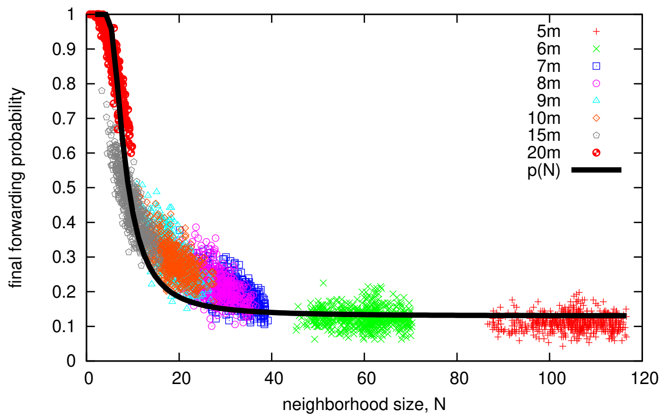


Fig. 2. The final forwarding ratio as a function of the neighborhood size for various network densities. Dots represent nodes and colors represent network density

where N denotes the number of neighbors of a node. In Fig. 2, this function corresponds to the solid black curve.

Given the above function denoting an optimal forwarding probability, $p(N)$, we modify GOSSIP3, such that each node autonomously assigns its forwarding probability p according to the function. To be precise, upon reception of a new data item, a node computes its current forwarding probability for the data item as $p_{curr} = p(N_{curr})$, where N_{curr} denotes the current number of neighbors of the node. Then, with the newly computed probability p_{curr}, the node decides whether to forward the data item, just like in GOSSIP3.

2) Estimating network density: For the above mechanism to work, each node is required to know not only how to compute $p(N)$, but also how many neighbors, N, it has, that is, how dense the network is in its vicinity. Due to the properties of wireless communication, notably signal attenuation and transmission collisions, accurately computing network density is a nontrivial problem. For this reason, we borrow from [11] the following heuristic solution.

We augment each packet broadcast by GOSSIP3 with an 8-bit sequence number. By analyzing sequence numbers in packets received from node j, node i can compute the packet reception rate $PRR(i, j)$. It can then use the computed packet reception rates for all nodes k in its radio range to obtain the estimate of the network density as follows:

$$N(i) = \sum_{k \in \{nodes\ in\ i's\ radio\ range\}} PRR(i, k)$$

This computation is redone periodically to account for changes in the node's vicinity since the last computation, for example, due to node mobility or fluctuations in the quality of wireless links. While not perfect, this simple heuristic for computing network density turns out sufficient for our algorithm.

B. Self-configured compensation mechanism

The second component of our self-configured algorithm is a novel mechanism for compensating for data that seem to be dying out. In the original GOSSIP3 protocol [2], if a node that decided not to forward a data item does not hear at least m neighbors forwarding this data item, it will forward the data item irrespective of its initial decision.

Although the authors of GOSSIP3 argue for using $m = 1$, our experiments provide evidence that this is not always the best configuration. In particular, in sparse networks $m = 1$ is simply insufficient, as only $m \geq 3$ provides the maximal coverage (cf. $\Delta = 20m$ in Fig. 1, top row). In denser networks, in contrast, $m = 3$ is an overkill as it increases the fraction of forwarders without any gain in coverage (cf. $\Delta = 10m$ in Fig. 1, top and middle row).

Furthermore, hearing m neighbors forward data does not necessarily mean that the data is safe, because not all neighbors are equal. More specifically, a neighbor that itself has few neighbors is less likely to create enough replicas of a data item to ensure that this data item does not die out than a neighbor that itself has many neighbors. Intuitively, in sparse regions of the network, the compensation mechanism should be triggered more aggressively than in denser regions. In other words, using just the number of neighbor retransmissions as a trigger for the compensation mechanism is too crude.

For these reasons, we propose a finer-grained heuristic for triggering the compensation mechanism. To this end, we make use of the fact that nodes already compute their neighborhood size, N. In addition, we require a node to embedd its value of N (up to 8 bits) in every packet it broadcasts.

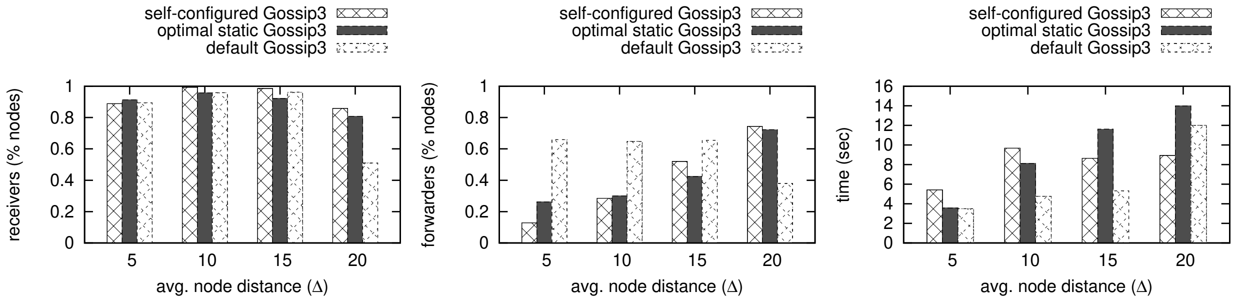
The heuristic works as follows. A node, i, that decided not to forward a data item waits for a predefined interval. At the end of the interval:

- If i heard no neighbor forward the data item, it triggers compensation: forwards the data item.
- If i heard at least 3 neighbors forward the data item, no compensation is necessary.
- Otherwise, i looks at its own neighborhood size, N_i, and the smallest neighborhood size, N_j, among the neighbors it heard forward the data item. If N_i or N_j is smaller than a connectivity threshold, N_{min}, node i decides to compensate and forward the data item; otherwise, it does not compensate.

Among others, the above heuristic ensures two important properties. A well connected node (one with $N \geq N_{min}$) does *not* trigger the compensation mechanism if the neighbors that took over forwarding are well connected or there are many of them. A poorly connected node, in turn, triggers the compensation mechanism unless many of its (few) neighbors take over forwarding. This heuristic minimizes the number of forwarders in dense network regions, while, at the same time, guarantees sufficient forwarding redundancy in sparse regions and on the borders between dense and sparse regions. The best performing value of $N_{min} = 7$ has been identified empirically.

C. Experimental results

In this section we compare our self-configured algorithm with (i) the best performing parameters of GOSSIP3, denoted as *optimal static* GOSSIP3, and (ii) the default parameters of GOSSIP3 as Haas et al. have suggested in the original paper [2], that are $p = 0.65$, $m = 1$ and $k = 1$ (denoted



(a) coverage (b) forwarders (c) latency

Fig. 3. Receivers, forwarders and latency for diverse densities. Impact of self-configured GOSSIP3 compared to static parameters: optimal and default GOSSIP3

as *default* GOSSIP3). For the self-configured and the optimal static GOSSIP3 $k = 0$. The remaining parameters for the optimal static configuration are in turn as follows:

	$\Delta = 5m$	$\Delta = 10m$	$\Delta = 15m$	$\Delta = 20m$
p	0.1	0.2	0.2	0.4
m	1	1	1	3

With respect to coverage our approach performs the same or better than the two static configurations as shown in Fig. 3a. As expected, the default parameters of GOSSIP3 perform poorly in sparse networks ($\Delta = 20m$), where coverage does not even reach 60% due to insufficient compensation ($m = 1$).

In Fig. 3b we see that our self-configured algorithm induces minimal traffic, intended as fraction of forwarders. In this respect, the default configuration of GOSSIP3 involves a high number of forwarders, which is way beyond what is required for reaching high coverage. For the default GOSSIP3 configuration the low fraction of forwarders when $\Delta = 20m$ is not to be considered, given the low resultant coverage. Our approach is only slightly outperformed by the optimal static GOSSIP3 for $\Delta = 15m$.

Latency is mainly determined by the number of hops and the extent to which compensation is done. This is reflected in Fig. 3c, where latency increases as the inter-node distance increases. Moreover, our self-configured approach reduces the latency for sparse networks such as $\Delta > 15m$, due to the sufficiently high initial probability.

V. CONCLUSIONS

In this paper we revisited the well-known GOSSIP3 protocol [2] for message dissemination in wireless ad-hoc networks.

We performed extensive experimental analysis in diverse network densities, and under utterly high network utilization, a setting not investigated for GOSSIP3 before. We observed that the protocol's input parameters are highly sensitive to the density of the network, and we explored this relationship, with respect to three metrics: dissemination coverage, traffic generated, and latency observed.

Such a dependency between the configuration parameters and the properties of the underying network may discourage the use of GOSSIP3 when the value of the parameter cannot be unambigiously determined, for example, in scenarios with mobility or unpredictable node density.

One of the major findings of this paper is a novel algorithm that alleviates this shortcoming of GOSSIP3. Our algorithm determines a node's rebroadcast probability, p, in such a way that the dissemination protocol can retain its optimal performance irrespectively of the density of the network it operates in.

This method results in a significant increase of the application range of GOSSIP3: the same configuration can operate optimally in sparse, dense, and heterogeneous networks, making the protocol more attractive for real-world deployments.

ACKNOWLEDGMENTS

This work was partially supported by the Foundation for Polish Science under grant HOMING PLUS/2010-2/4, co-financed from the Regional Development Fund of the European Union within the Innovative Economy Operational Program, and a START scholarship (K. Iwanicki).

The authors would like to thank Dr. Rena Bakhshi for her valuable help with the curve-fitting technique.

REFERENCES

[1] A. Gaba, S. Voulgaris, and M. van Steen, "Group monitoring in mobile ad-hoc networks," in *E-Democracy*, 2010.

[2] Z. Haas, J. Halpern, and L. Li, "Gossip-based ad hoc routing," in *INFOCOM*, 2002.

[3] W. Peng and X.-C. Lu, "On the reduction of broadcast redundancy in mobile ad hoc networks," in *MobiHoc*, 2000.

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

[5] B. Williams and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks," in *MobiHoc*, 2002.

[6] Z. J. Haas and M. V. Nikolov, "Towards optimal broadcast in wireless networks," in *MSWiM*, 2011.

[7] Q. Zhang and D. Agrawal, "Dynamic probabilistic broadcasting in MANETs," *J. Par. & Dist. Comput.*, vol. 65, no. 2, pp. 220 – 233, 2005.

[8] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *MobiCom*, 1999.

[9] S. Pleisch, M. Balakrishnan, K. Birman, and R. van Renesse, "Mistral: efficient flooding in mobile ad-hoc networks," in *MobiHoc*, 2006.

[10] C. Ellis, H. Miranda, and F. Taïani, "Count on me: lightweight ad-hoc broadcasting in heterogeneous topologies," in *M-PAC*, 2009.

[11] B. Hull, K. Jamieson, and H. Balakrishnan, "Mitigating congestion in wireless sensor networks," in *SenSys*, 2004.