Towards Congestion-Aware All-to-All Information Dissemination in Mobile Ad-Hoc Networks

Albana Gaba Spyros Voulgaris Maarten van Steen

Dept. of Computer Science, VU University Amsterdam, The Netherlands
{agaba,spyros,steen}@cs.vu.nl

ABSTRACT

We discuss protocols for all-to-all dissemination in ad-hoc wireless networks. We identify shortcomings and challenges related to high traffic and explore their causes by observing the performance at the MAC layer. Through simulations we show that a congestion assessment mechanism can reduce more than 1/4 of the traffic on a good part of the nodes without affecting dissemination performance. We conclude that MAC-layer congestion awareness is important for improving application-level efficiency, and that we should not separate the two when dealing with all-to-all broadcasting.

1. INTRODUCTION

As wireless communication is becoming increasingly ubiquitous, we see more and more applications realized by means of networks of battery-powered wireless devices. A large class is formed by traditional applications of wireless sensor networks, whereas novel ones encompass areas of ambient intelligence and social networking. Moreover, as we are gradually mastering fundamental techniques for routing and energy consumption, these distributed applications are becoming more extreme in terms of the number of nodes they can support, but also with regards to the resource constraints imposed on the nodes.

We are seeking solutions for what we refer to as extreme wireless distributed systems: systems consisting of hundreds, perhaps even thousands of wireless devices, each constrained in terms of compute power and memory, and often equipped with only a relatively small, low-capacity battery. We assume that nodes can easily join and leave the network, and can also easily change their relative position in the network. As a consequence, we can no longer rely on traditional routing protocols, as by the time a route has been discovered the changes in the network will have rendered that route useless.

Under these assumptions, a fundamental building block is to disseminate information across the network and to allow a node to take application-specific decisions based on the information it has received so far. To make matters concrete, consider a scenario in which a family or similar group of people attends a large social event. To keep track of each others presence, each group member regularly broadcasts presence information. To prevent maliciously intended traffic analysis for tracking of a lost or isolated member, presence information is encrypted and flooded through the network to be recognized only by members of the same group [3].

In this scenario, all-to-all information dissemination is crucial. Considering that we require any broadcasting scheme to be able to handle near-continuous changes in the underlying network, gossip-based dissemination seems to be most promising. Unfortunately, because of the limited communication capacity of the network, it is unclear whether current solutions can be effectively adopted. In this context, we study a popular protocol, called GOSSIP3, as originally proposed by Haas et al. [4].

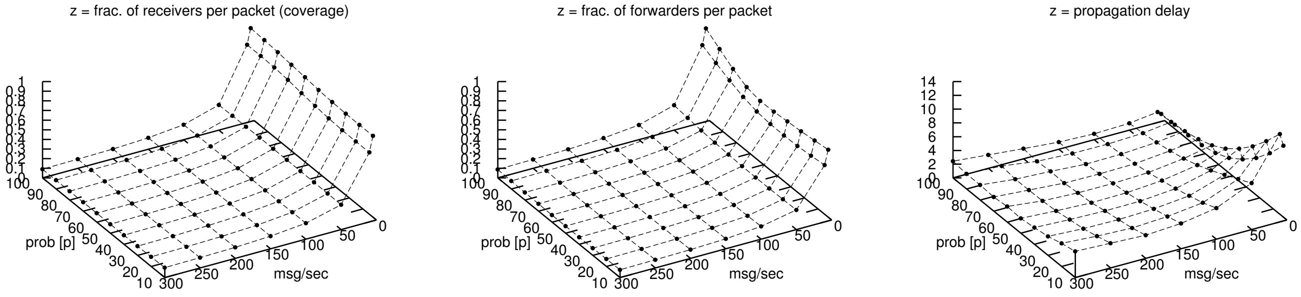
In particular, we are interested to see under which circumstances an adaptation of GOSSIP3 can suffice to provide efficient all-to-all broadcasting. To this end, we assume that the underlying network makes use of a CSMA MAC layer and investigate under different message generation rates the extent that a message is successfully broadcast to all nodes in the network. We observe that GOSSIP3 will quickly lead to severe network congestion, but not uniformly distributed across all nodes.

Our main contribution is that we show that it is necessary to make application-level dissemination protocols such as GOSSIP3 much more congestion aware than is currently the case. We advocate that MAC-layer congestion information needs to be taken into account in order to prevent severe performance degradation, and thus that only application-level solutions are not sufficient.

2. BROADCAST PROTOCOLS

The main problem with broadcast protocols is to reach as many nodes as possible in a reasonable time while having low message overhead. A common approach has been to simply flood messages, but flooding has been shown to generate lots of network traffic resulting in congestion and high packet loss rates[1]. As it turns out, carefully deciding when a node should relay a packet is fundamental to achieve good performance. Popular techniques are the following:

[1] We use the term “message” for application-level traffic and “packet” for MAC-layer traffic.



(a) Coverage per packet (b) Rebroadcasts per packet (c) Average latency

Figure 1: Application level performance for different values of p and message generation rates.

Probability-based: Upon receiving a message, a node retransmits it with a certain probability, which is assigned statically [4] or dynamically [12; 16]. A dynamic probability allows nodes in dense areas to retransmit with a lower probability compared to nodes located in sparse areas.

Counter-based: This technique is based on the premise that retransmitting a message that has already been received multiple times bears little added effect [9]. When a message is received for the first time a counter is started, and a *random assessment delay* (RAD) is set between 0 and T_{max} seconds. During the RAD period, the counter is incremented for each redundant message received after which it is rebroadcast only if its counter is lower than a predetermined threshold.

Neighbor-knowledge-based: This technique requires nodes to build an overlay network based on the information received about nearby nodes [10; 11]. The basic idea is to allow only the highest connected nodes to act as relays. To this end, a node maintains up-to-date neighborhood information by periodically exchanging presence messages.

Area-based: In this case, a node decides to rebroadcast a message based on its distance from the sending neighbor [9]. The idea is that the efficacy of a rebroadcast depends on the additional area that a relaying node might cover compared to that of the sender. Messages received from far-away neighbors should be rebroadcast, those from close-by neighbors should not.

The first two techniques are the simplest as they rely only on locally collected data and do not require additional equipment, such as GPS. Protocols requiring neighborhood information impose more communication due to presence messages. In addition, in case of high packet loss rates or mobility, such information may be quickly outdated, leading to even higher communication demands.

The protocol we adopt for this paper is GOSSIP3 [4], which is a combination of a probability-based and counter-based technique. Each node, upon reception of a message, decides with a certain probability p whether to rebroadcast or not. If it decides not to rebroadcast, the message is buffered for a random assessment delay (RAD) period between 0 and T_{max}. After that, if the message has not been received more than m times, it is rebroadcast.

We are interested to see whether GOSSIP3 can perform well as an all-to-all dissemination means in relatively extreme cases. In particular, we consider large networks in which new messages are injected at relatively high frequencies. In our experiments we evaluate per message (1) the average number of receivers (i.e., *coverage*), (2) the average propagation delay, and (3) the average number of rebroadcasting nodes.

3. EXPERIMENTAL SETUP

We run simulations using the OMNeT++ Mobility Framework [1] simulation environment. We consider a network of 300 nodes placed uniformly at random in a $900 \times 900 m^2$ area. We fix this random topology across all our experiments. The average node degree is 13.

Each node inserts 60 messages in the network at various periods. At the beginning nodes initiate the broadcasting at a random time between 0 and a given period and continue to send messages at a constant bit rate.

Furthermore, we deploy the standard IEEE 802.15.4 implementation which includes a CC2420 tranceiver and a CSMA MAC layer. Considering that our target devices are restricted in size and computation power, the 802.15.4 standard seems to fit our requirements well. The communication between nodes is determined by their transmission power and vicinity. At the physical layer, the Signal to Noise Ratio (SNR) modulation and Forward Error Correction (FEC) encoding/decoding functions define the bit error rate and, therefore, the throughput of the system. Note that OMNeT's low-level radio models have been validated, which is essential for the validity of our entire approach.

We introduce a small random delay (jitter) when scheduling a transmission just after packet reception. This helps to prevent massive collisions when several nodes in the neighborhood decide to simultaneously retransmit a just received message.

4. APPLICATION LAYER PERSPECTIVE

In this section, we limit our findings to what can be observed by looking at the application layer alone. In the following section, we dive into the MAC layer.

Our goal is to study broadcast protocols for all-to-all dissemination. As such, we are interested in the behavior of such protocols in the face of high traffic. We start with studying the behavior of GOSSIP3 for different levels of traffic and different values of the transmission probability p. We fix the parameter m to 1 which is the most appropriate value suggested in [4].

Fig. 1(a) plots the percentage of nodes that *receive* a given message, that is, the average coverage achieved. We notice that coverage is strongly dependent on the message generation rate (i.e., the traffic in the network), while the influence of the transmission probability p is hardly observable.

This should not come as a surprise. On the one hand, increased traffic pushes network capacity to its limits. When a large number of messages compete for finite network resources, it comes as a natural consequence that their dissemination is limited.

On the other hand, lowering p does not really affect coverage. Even when a node receives a message and chooses *not* to initially rebroadcast it, it may still do so later on. As a result, when not hindered by excessive traffic, messages manage to eventually spread to the whole network. When network traffic becomes a bottleneck, however, the effect is still the same, irrespective of the value of p.

Fig. 1(b) shows the percentage of nodes that on average *rebroadcast* a message. The initial observation is that this graph resembles the one in Fig. 1(a). This is to be expected, especially for high traffic. When network congestion is high and coverage is as low as 10%, a node receiving a message will most likely be the only one receiving it among its neighbors, and will thus end up rebroadcasting it as well.

A closer look at Fig. 1(b), however, reveals that at low traffic levels probability p affects the number of nodes that rebroadcast a message. As expected we have a maximum for $p = 1$, as *everyone* will be rebroadcasting every message, while the percentage of relaying nodes drops to a minimum for $p = 0.6$.

Finally, Fig. 1(c) shows the influence of traffic and p on the average propagation time. As expected, low probability p increases the overall propagation time, as nodes do not always rebroadcast messages instantly but only after the RAD period.

Given that $p = 0.6$ achieves low dissemination latency with relatively fewer rebroadcasts, from now on we fix p to this value. This is in accordance with [4], where a value of $p = 0.65$ is considered to be the optimal.

These graphs, nevertheless, show that in the face of congestion, broadcast algorithms like GOSSIP3 have an immense difficulty in disseminating messages, despite the regulatory compensation technique. Even worse, the compensation technique of GOSSIP3 turns out to be *worsening* the problem in the presence of congestion.

Fig. 2 shows the percentage of received messages a node rebroadcasts, as a function of the message generation rate. We see that the higher the traffic, the higher the percentage of messages that are rebroadcast by a node. It is not hard to see that when packets are being lost, as explained above, a node may mistakenly assume that a received message has not been delivered to any of its neighbors, thus opting to rebroadcast it, but this time redundantly.

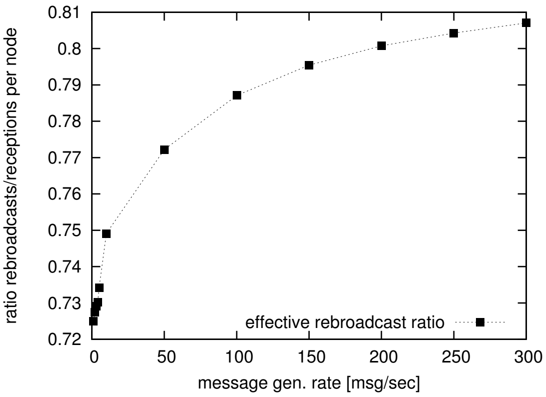


Figure 2: Effective p: As network stress increases, GOSSIP3 tries to compensate for packet drops by increasing the probability to rebroadcast received messages, recursively increasing congestion.

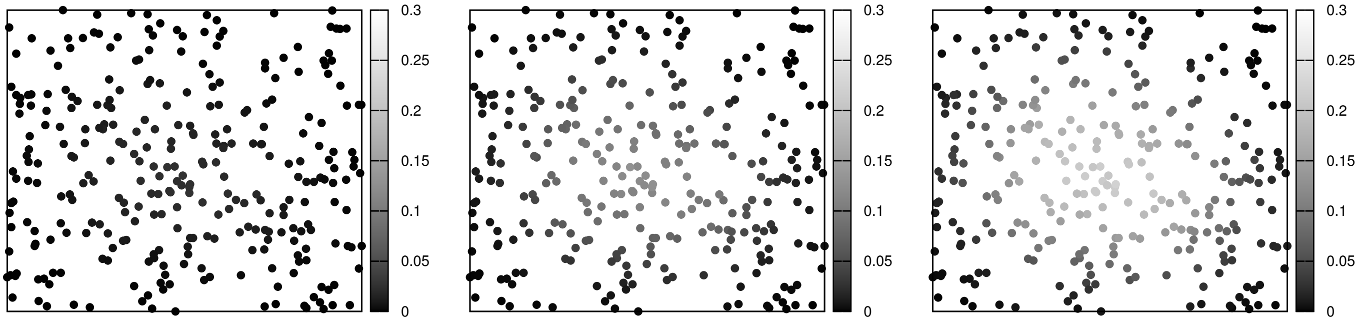
This is a key observation, as it leads us to the intermediate conclusion that trying to push one's messages may in fact have countering results in the face of congestion. This hints at the direction we take in the following section, where we study congestion at the MAC layer, and look at a solution that may improve matters.

5. MAC LAYER PERSPECTIVE

The Media Access Control (MAC) layer is responsible for regulating node access to a shared communication channel. In our experiments we consider the IEEE 802.15.4 standard MAC protocol, which is designed for low bit rates and low-power communication applications. It consists of a contention-based CSMA-CA protocol that requires nodes to sense a channel before packet transmission. Time is divided into slots. Before transmitting a packet, the sender checks if the medium is idle. In that case, the sender waits for the next time slot and, if the medium is still idle, it assumes it has won the contention and transmits the packet. Otherwise, if the medium is busy, it increases a backoff counter for that packet and schedules a new attempt after a random number of slots. This number is chosen at random between 0 and $2^{BE} - 1$, where BE is a backoff exponent having a protocol-specific initial value (we use the default value 3) and is incremented after every backoff. If a packet fails to be transmitted after five backoffs, it is dropped.

To assess the behavior of the MAC layer, Fig. 3 plots the number of packets that have been *transmitted* and the number of packets that have been *dropped* by each node, as a function of the message generation rate. Note that the total number of messages generated in each experiment is the same, but they are generated at different rates.

By pushing new messages at a higher frequency, we see that the number of packets transmitted by the MAC layer de-



(a) 5 mps (b) 10 mps (c) 20 mps

Figure 4: Packet drop rate per node for $p = 0.60$ represented by the intensity of the dots. As the frequency of message generation increases, so does the number of dropped packets for central nodes.

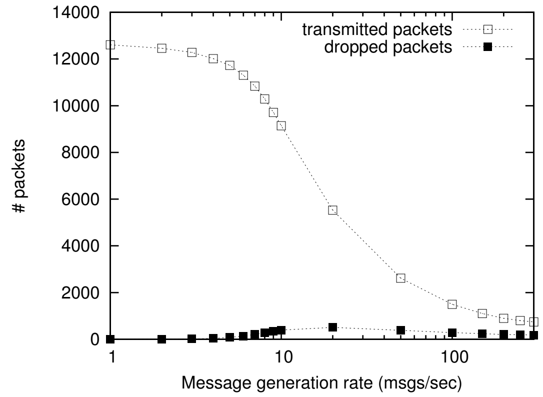


Figure 3: Number of packets transmitted and dropped per node, for $p = 60$ (x-axis in log scale)

creases significantly. This is a result of multiple factors stemming from high congestion, manifesting themselves as follows:

- **Hindered transmission** High utilization of the medium forces the MAC layer to drop packets after it has failed to occupy a free slot for a number of attempts. This lowers the *transmission success ratio*, that is, the ratio of the number of packets a node manages to broadcast, over the total number of packets handed to its MAC layer.
- **Hindered reception** Even when the MAC layer manages to broadcast a packet, the chance of a collision is increased due to channel saturation. This lowers the *reception success ratio*, that is, the number of packets correctly received by a node, over the total number of packets that it should receive in the absence of collisions (e.g., at very low traffic).
- **Application issues** Multihop message dissemination is inherently exponential in the number of broadcasted packets. In effect, the loss of a single packet at the early hops of a message's dissemination will potentially prune a whole tree of packet broadcasts. This has a multiplicative effect on the overall decrease of broadcasted packets.

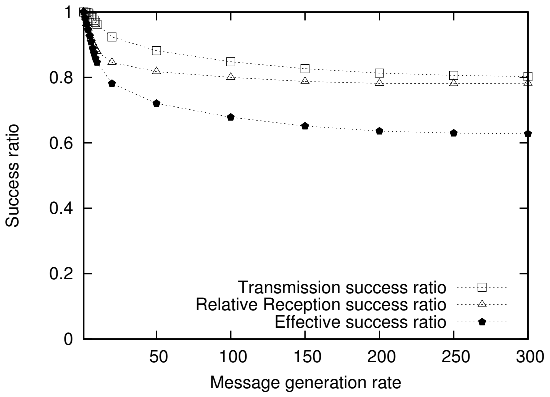


Figure 5: Transmission and reception success ratios as a function of messages generated per second, relative to the respective success ratios for 1 msg/sec, for different message generation rates and $p = 0.60$.

Fig. 5 plots the transmission and reception success ratios, for our simulated spectrum of message generation rates, relative to the respective success ratios at the lowest traffic we considered (1 msg/sec).

Now we investigate the correlation between the location of a node and the performance of its MAC layer. To this end, Fig. 4 depicts the nodes at their coordinates in the simulated topology, color coded based on their respective packet drop ratio (i.e., the complement of the transmission success ratio), for 5, 10, and 20 msg/sec. Interestingly, we see that performance degrades more rapidly for central nodes. This nonuniform performance degradation occurs despite the uniform distribution of the message-generating nodes across the network and the unbiased broadcasting nature of GOSSIP3. This suggests that central nodes are subject to higher load compared to the nodes in the periphery.

To quantify the packet drop rate as a function of node location, we group nodes according to their *eccentricity*, a metric borrowed from graph theory. For wireless networks, we compute the eccentricity of a node as the number of steps it takes an expanding ring protocol initiated at that node to cover the whole network. Clearly, nodes in the center have minimum eccentricity, while nodes at the periphery have a maximum eccentricity.

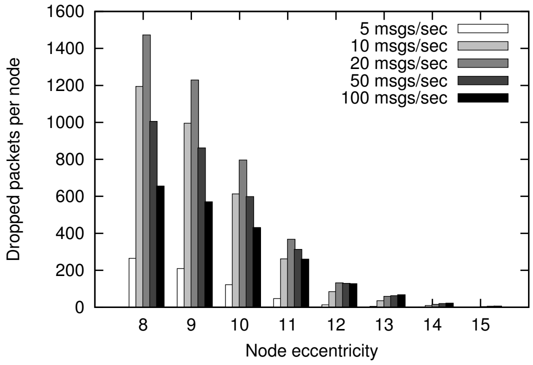


Figure 6: Dropped packets at the MAC layer for $p = 0.60$ and different message generation rates.

Fig. 6 shows the number of packets dropped per node for various message generation rates, grouped by node eccentricities. The significantly higher load in the center of the network (low eccentricity values) is clearly visible. Another interesting observation is that for any given eccentricity, a message generation rate of 20 msg/sec maximizes the number of packets dropped, while the number of dropped packets diminishes both for lower and for higher message generation rates. For lower rates, the explanation is obvious: the network is not used at its full capacity. For higher rates, however, the number of dropped packets is lower simply because the total number of packets is significantly lower (see Fig. 3).

6. CONGESTION-AWARENESS

The results emerging from the previous sections lead to the following two key observations:

- Congestion is more severe in the center of the network, as seen in Fig. 6.
- GOSSIP3 is highly sensitive to congestion, perceived as high packet loss ratio. Indeed, high packet loss ratio is interpreted by the application layer as lack of redundancy in message propagation, which causes more rebroadcasts, aggravating traffic load, as seen in Fig. 2.

The above findings suggest that if nodes can individually detect congestion at their MAC layer, they can instruct their application layer to adapt accordingly. More specifically, in the presence of congestion, the application layer should refrain from rebroadcasting, contrary to how the original GOSSIP3 protocol behaves.

To this end, we propose to change GOSSIP3, by letting a node decide to rebroadcast a message for redundancy compensation with probability q, rather than deterministically. Of course, q will be dependent on local congestion.

We apply this adjustment to nodes at the center of the network, having eccentricity 8 or 9. Fig. 7 and Fig. 8 show

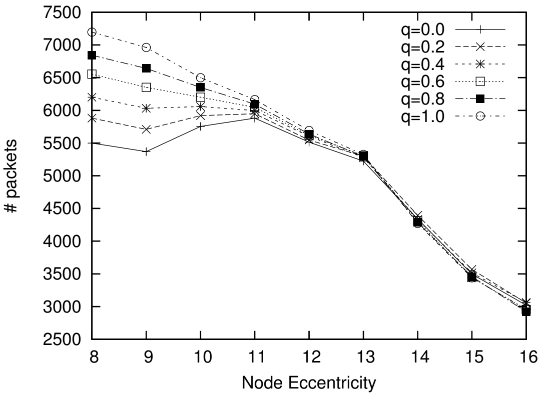


Figure 7: Number of rebroadcasts per node for $p = 60$ and message generation rate 20 msg/sec. Lines plot different values of compensation probability q.

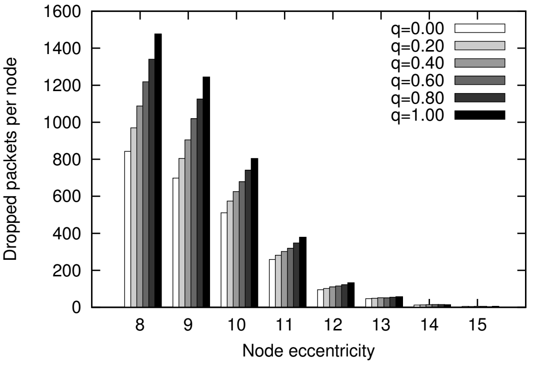


Figure 8: Dropped packets for $p = 0.60$ and message generation rate 20 msg/sec, for different values of compensation probability q. Only nodes with eccentricity 8 or 9 apply this probability q for transmitting in the compensation phase. Note that $q = 0$ corresponds to probabilistic flooding whereas $q = 1$ corresponds to the original GOSSIP3.

respectively the number of rebroadcasted packets per node (from the application layer) and the number of dropped packets per node (at the MAC layer). For q = 1, we have the original GOSSIP3 protocol, while for q = 0 central nodes completely abstain from the compensation phase, corresponding to probabilistic broadcasting. We observe that lowering q to 0 decreases congestion significantly for central nodes. However, coverage is hardly affected, as can be witnessed from Fig. 9, where we plot coverage as a function of probability q for different message generation rates. As it turns out, coverage essentially stays the same while changing q for central nodes.

The above findings imply that about 25% of rebroadcasts done by the central nodes are ineffective. Such redundancy has a negative impact on the performance of the MAC layer which reaches 40% increase in the number dropped packets as suggested by Fig. 8. We notice that under congestion the benefits offered by GOSSIP3 are insignificant compared to

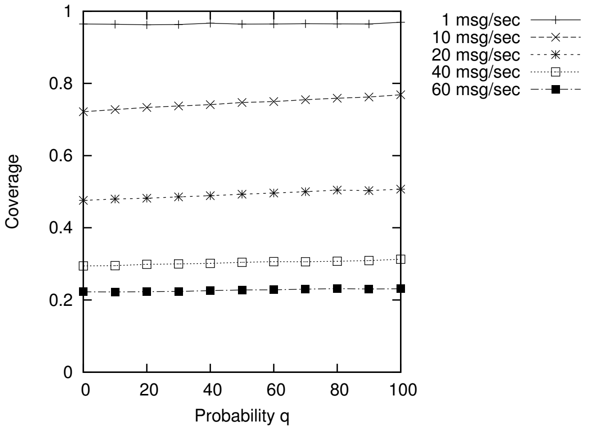


Figure 9: Effect of compensation probability q on overall coverage, for $p = 0.60$ and message generation rate 20 msg/sec.

the induced communication overhead. Indeed, in this case a simple probabilistic broadcasting can achieve almost the same performance and generate 25% less rebroadcasts.

We believe that in presence of congestion, data dissemination protocols whose forwarding policy is based on the received messages may manifest the same issues. Therefore, information from the MAC layer would benefit such protocols by letting the application take informed decisions about rebroadcasting based on the load of the network.

7. RELATED WORK

Using MAC-layer congestion information to improve higher level protocols has been studied quite extensively, notably for 802.11 networks [6]. Several studies on routing protocols in mobile ad hoc networks take congestion into account in order to select routes (see, e.g., [8; 13]). Also, several studies consider the relation between MAC-layer congestion and adopting TCP for MANETs [5; 7]. In many cases, congestion awareness is used to support single-path routing algorithms, leading to sophisticated scheduling algorithms that cannot be easily deployed for the type of networks we consider here.

Broadcast protocols have been extensively studied in [14]. Despite the relatively small network size they use for the experiments, they show the performance degradation for high message generation rates. Experiments suggest suitable parameters settings for a counter-based protocol in dense areas, but they do not consider MAC-level congestion information. In general, we see that cross-layer optimizations barely touch upon gossip-based dissemination schemes [2]. In addition, it has been recognized that constructing effective congestion detection mechanisms is not obvious [15].

By-and-large, despite the large body of work on cross-layer designs and deploying MAC-layer congestion information, we see that only few studies in MANETs address congestion-aware dissemination at the application level.

8. CONCLUSIONS AND FUTURE WORK

In this paper we evaluated the performance of the GOSSIP3 dissemination protocol under varying network loads. Data collected from the MAC layer reveals useful insight about the distribution of the load in a uniformly distributed network. We show that there is a correlation between a node's centrality and the imposed traffic load, despite the uniform broadcast nature of the dissemination protocol. In the presence of congestion, a high rate of dropped packets has a negative impact on the decisions of the application layer, which tries to forward packets more aggressively, further worsening the congested network.

The main conclusion from this research is that MAC layer information is *needed* by the application to take informed decisions on message forwarding. In other words, cross-layer design seems to be inevitable in all-to-all broadcasting. How to effectively establish this for improving coverage and further reduction of redundant messages will be our next step.

References

[1] Mobility Framework (MF) for simulating wireless and mobile networks using OMNeT++. `http://mobility-fw.sourceforge.net/`.

[2] M. D. Felice. "*Cross-Layer Optimizations in Multi-Hop Ad Hoc Networks.*" PhD thesis, U. Bologna, 2008.

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

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

[5] G. Holland and N. Vaidya. "Analysis of TCP Performance over Mobile Ad Hoc Networks." *Wireless Networks*, 8:275–288, 2002.

[6] Y.-C. Hu and D. Johnson. "Exploiting congestion information in network and higher layer protocols in multihop wireless ad hoc networks." In *Proc. ICDCS*, 2004.

[7] D. Kliazovich and F. Granelli. "Cross-layer Congestion Control in Ad Hoc Wireless Networks." *Ad Hoc Networks*, 4(6):687–708, 2006.

[8] Y. Lu, W. Wang, Y. Zhong, and B. Bhargava. "Study of distance vector routing protocols for mobile ad hoc networks." In *Proc. PerCom*, 2003.

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

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

[11] W. Peng and X.-C. Lu. "Ahbp: An efficient broadcast protocol for mobile ad hoc networks." *J. Comp. Sc. & Techn.*, 16(2):97–102, 2001.

[12] D. Scott. "Dynamic probabilistic retransmission in ad hoc networks." In *Proc. ICWN*, 2004.

[13] D. Tran and H. Raghavendra. "Congestion adaptive routing in mobile ad hoc networks." *IEEE Trans. Par. & Dist. Sys.*, 17:1294–1305, 2006.

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

[15] W. Wu, Z. Zhang, X. Sha, and C. He. "Auto Rate MAC Protocol based on Congestion Detection for Wireless Ad Hoc Networks." *Inform. Techn. J.*, 8(8):1205–1212, 2009.

[16] Q. Zhang and D. Agrawal. "Dynamic probabilistic broadcasting in MANETs." *J. Par. & Dist. Comput.*, 65(2):220 – 233, 2005.