A Study of Adaptive Gossip Routing in Wireless Mesh Networks

Bastian Blywis, Mesut Güneş, Sebastian Hofmann, and Felix Juraschek

Institute of Computer Science - Distributed Embedded Systems
Freie Universität Berlin, Germany
`{blywis,guenes,shof,jurasch}@inf.fu-berlin.de`

Abstract. Gossip routing is an approach to limit the overhead of flooding in wireless networks. Each node, that receives a packet that would normally be flooded, applies a probabilistic approach. The packet is either forwarded with probability p or dropped with $1 - p$. This paper is a follow-up to our last study that evaluated the approaches by Haas et al. in the DES-Testbed, a wireless mesh network. Four different gossip routing variants were discussed which used static parameters to determine the value p. In this study, we compare the proposals by four other entities and discuss whether they show an overall improvement regarding the two most important metrics in this domain: reachability and redundancy. We also discuss the assumptions and parameters of the simulation studies in the context of our experiments in a real world deployment.

Keywords: Wireless Mesh Network (WMN), Gossip Routing, Probabilistic Flooding, Testbed.

1 Motivation

Gossip routing, also called gossiping or probabilistic flooding, is an approach to improve the deficiencies of flooding. Flooding is an important service in wireless networks that is used when a source node has to send data to all nodes in the network. This is usually achieved by collaborative broadcasting. Each node, a router, that receives a packet forwards it by re-broadcasting the packet. In this way the packet traverses the whole network. Flooding in wireless networks is additionally "aided" by the physical layer broadcast medium and requires routers to send only one copy instead of one packet per dedicated link in wired networks.

Routing protocols use flooding to spread topology information or route requests and replies in the network. *Link state routing* protocols, that are a representative of the class of proactive protocols, flood neighborhood information. For example, the *Open Shortest Path First* (OSPF) protocol [1] uses flooding based on multicast addressing. Reactive protocols execute an ad-hoc route discovery process when a packet has to be sent to some destination and no routing information is available. In many reactive protocols, a route discovery message is sent to all neighbors. If they cannot provide the required routing information, the packet is broadcasted to their neighbors and thus often flooded over the

J. Zheng, D. Simplot-Ryl, V.C.M. Leung (Eds.): ADHOCNETS 2010, LNICST 49, pp. 98–113, 2010.

whole network. If a node can eventually answer with a route reply, the corresponding packet is either sent as unicast or flooded again in the reverse direction, depending on the particular protocol.

Although flooding is very simple to implement, it has some deficiencies. As all nodes broadcast received packets, routers in the same broadcast domain may receive the same packet multiple times. Redundant data is sent over the medium. Especially in wireless networks, broadcast transmission may only be possible with low data rates, e.g., the `BSSBasicRateSet` data rates defined in IEEE 802.11 [2]. These packets can occupy significant time on the medium, which curbs high-speed unicast data transmissions. Further on, they also increase the contention for the medium access and the overall noise level. Due to these issues and the generally unreliable wireless medium, several packets might be lost. When we consider that a route request is lost, the result could be a sub-optimal or failed route discovery. The standard acknowledgment-based loss detection cannot be applied in flooding, as the neighbor set is either not known or the sender would experience the ACK implosion problem [3] leading to increased contention and in some cases could overwhelm the node. If the links are unidirectional, acknowledgments might not even be possible. It cannot be guaranteed that all nodes in the network receive the packets sent by a source. The optimization of the flooding service is therefore an important task for the optimal operation of wireless networks.

To minimize the redundancy while still retaining full reachability, *gossip routing* can be used in place of flooding. In contrast to flooding, where packets are usually forwarded as long as the *time to live* (TTL) value in the packet header does not reach zero, the approach is probabilistic. Each node forwards packets with a given probability p or drops them with probability $1 - p$ in the most simple variant. Gossip routing is related to *epidemic routing* [4] where messages are stored on (mobile) nodes and are probabilistically forwarded to neighbors applying a "store-carry-forward" paradigm which models the spreading of diseases in a population.

In our previous study [5] we discussed and evaluated three gossip routing variants by Haas et al. [6,7] and one of ours. The experiments were run in the *Distributed Embedded Systems - Testbed* (DES-Testbed) at *Freie Universität Berlin*, a wireless mesh network (WMN) [8]. Our results showed that gossip routing can reduce the redundancy in real world networks by up to 26% compared to flooding while at the same time a larger number of nodes is reached from the source node. Further on, we showed that the position of the source node has a crucial influence on the overall performance and which of the variants performs better. In contrast to the simulation-based study by Haas et al., the number of nodes that received packets from the source node was limited. Although each node received some packets from the source node during the whole experiment, there was no case when all nodes received the same packet together, even when we used $p = 1.0$ to achieve flooding. The properties of our real world networks significantly deviate from the models that are applied in simulations. This fact

Algorithm 1. PCBR - *gossip7*

```
Require: received packet with $id$
  if $id$ not in $list_{rcvd}$ then
    $c_{id} = 1$
    insert $id$ in $list_{rcvd}$
    set RAD $\in (0, T_{max}]$ and wait for RAD to expire
    if $c_{id} \leq m$ then
      forward packet with probability $p$
    else
      remove $id$ from received list
    end if
  else
    if $c_{id} \leq m$ then
      $c_{id} = c_{id} + 1$
    else
      remove $id$ from $list_{rcvd}$, stop RAD
    end if
  end if
```

underlines the necessity of further sophisticated experimentally driven studies. There is currently limited experience with gossip routing in real world networks.

In this study, we use the same experiment setup to evaluate four other gossip routing variants. Major differences are that the forwarding probability of packets is not configured statically, network-wide and that the algorithms are more adaptive to the network topology. We evaluate the measured data considering two of the most important metrics in this domain. Gossip routing shall ensure that all nodes in the network receive the packets sent by a source. We deem the *reachability* as prime objective as all other optimizations are basically void if we reach only a limited subset of nodes. The *redundancy* has to be reduced so that the medium can be used for data flows with higher data rates.

The remainder of this paper is structured as follows. In Section 2 the related work is discussed and the used gossip routing variants are introduced. The experiment setup and the implementation based on our routing framework are described in Section 3. All measurements are evaluated and discussed in Section 4. The paper ends with a conclusion in Section 5.

2 Related Work

The following gossip routing variants are part of this study. The numbering of the algorithms is continued from our previous publication were we studied *gossip0* to *gossip5*. As *gossip3* showed good performance and because it is used as basis for two other variants, it has also been included in this study for comparison.

Algorithm 2. P-AODV - *gossip8*

```
Require: received packet for the first time
Require: number of neighbors $n > 0$
  $p_{max} = 0.9$
  $p_{min} = 0.4$
  $p = \max(p_{max}(\frac{p_{max}^{n-1} - p_{max}^{n}}{1 - p_{max}}), p_{min})$
  if $random(0, 1) < p$ then
    forward packet
  end if
```

Algorithm 3. DPR - *gossip9*; n and n_c represent the number of elements in the particular sets

```
Require: received packet for the first time
  if $n \leq n_f$ then
    $p = 1$
  else
    $p = 1 - e^{-\frac{n - n_c}{n}}$
  end if
  if $random(0, 1) < p$ then
    forward packet
  end if
```

2.1 gossip3

Haas et al. propose multiple gossip routing variants that support an initial flooding for k hops [6,7]. This approach shall ensure that the gossiping does not terminate early on when there are few neighbors to forward the packets. The authors argue that this is especially important to reduce border effects when the source node is near the edge of the network. The *gossip3* variant additionally considers random networks with varying node degree. When a node would normally drop a packet due to an unlucky random draw, it is stored instead. If fewer than m duplicates are received in a specific time, the packet is broadcasted nevertheless; otherwise it is finally discarded. The value of the parameter m is configured statically and is the same for all nodes in the network. Experiments in a static, random network showed that a threshold of about $p = 0.65$ is sufficient for most nodes in the network to receive the packets from a source node. Another experiment scenario with a mobile network applied *gossip3* for the route discovery phase of AODV.

2.2 gossip6

Like Haas et al., Shi and Shen try to improve the route discovery phase of AODV. Their approach is called *Adaptive Gossip-based Ad Hoc Routing* (AGAR) [9] and is based on *gossip3*. When a packet is received and the random number is above the threshold p, the packet is stored. Instead of always broadcasting the packet when fewer than m duplicates have been received in the waiting time, the following formula is used to determine whether the packet shall be sent: $random(0, 1) \leq p/(d + 1)$, where d is the number of received duplicates. The authors argue that this adaptive modification reduces flooding in networks with a high node degree. The second chance to forward packets is smoothly decreased leading to a graceful degradation.

2.3 gossip7

Mohammed et al. propose a *Probabilistic Counter-based Route discovery* (PCBR) for AODV [10]. The general idea is that packets are always stored when they are received and the number of duplicates is counted. Packets are forwarded after the *random assessment delay* (RAD) timer has expired and when not enough duplicates ($d \leq m$) were received. The RAD timer value is randomly chosen from the interval $(0, T_{max}]$, see Algorithm 1. The basic approach is the same as in *gossip3*, but packets are always stored and the initial k hop flooding is missing. $T_{max} = 10\ ms$ was used in experiments by Mohammed et al. to achieve a low delay.

2.4 gossip8

Hanashi et al. propose P-AODV [11]. When a broadcast packet (AODV route request) is received for the first time by a node, the probability p to forward the packet is calculated based on the number of neighbors n. p has a high value when there are few neighbors and has a low value when the node degree is high. The probability p is additionally bounded by p_{min} and p_{max}. A simplified but complete representation of the original algorithm is shown in Algorithm 2. The approach is similar to *gossip2* by Haas et al. where a different but static probability p_2 is used when the node degree is low.

2.5 gossip9

Jamal-Deen Abdulai proposes the *Dynamic Probabilistic Route* (DPR) algorithm [12] for AODV. When the node degree n is lower than or equal to n_f, the algorithm turns to flooding. n_f is the average node degree. When the node degree n is higher than n_f, the additional coverage that can be achieved by forwarding the packet determines the value of p. Each node that forwards a packet attaches a list of its neighbors n_c. Upon reception of a packet, the node determines the set of his neighbors that (probably) have not yet received this packet. If the set is empty, the node will not forward the packet; otherwise the larger the set, the higher the forwarding probability. The approach is simple since it requires only 1-hop neighborhood information to apply the self-pruning. The pseudo code is shown in Algorithm 3.

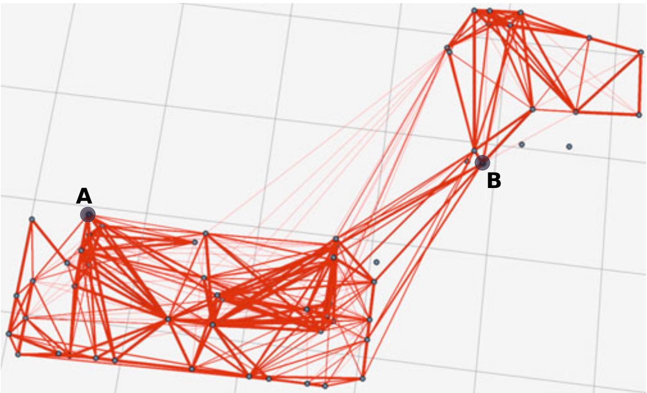


Fig. 1. Overview of the locations of the nodes including the link quality as determined by a daemon that implements a HELLO protocol using the ETX metric. Thicker lines represent a better ETX value and a higher quality link. The figure shows links on channel 13. Source: [13].

2.6 Focus of This Study

Significant focus in the domain of gossip routing has been on the route discovery phase of AODV. Our experiments as specified in Section 3 focus on the pure performance of gossip routing. In the following we will always refer to the algorithms by their *gossipX* names, to emphasize that we are focusing only on the gossip routing and not the application in the AODV protocol. The goal is to research probabilistic flooding as a service for any routing protocol.

3 Experiment Setup

All experiments have been run in the DES-Testbed, a hybrid multi-transceiver wireless mesh and sensor network testbed [8]. 59 wireless mesh routers deployed in two adjacent buildings were used. The larger building had 42 and the smaller building 17 nodes which are distributed over multiple levels. Both buildings are close to each other and there are normally multiple links in-between. Figure 1 shows the topology measured with a daemon [13] that implements the ETX metric [14]. Despite the shown snapshot, the testbed is not partitioned but some links have a low quality. From our last experiment batch we know that all nodes in the network received some packets. Experiments were only run at night and on weekends when there are no or very few people or sources of interference in the vicinity. The average node degree of the network is about 7 with a maximum of 23. Each router is equipped with three IEEE 802.11a/b/g network cards. We used one card configured to channel 13 for the experiments as it not used by the campus-wide WLAN.

All variants of the gossip routing protocol were implemented based on the *Distributed Embedded Systems - Simple and Extensible Routing-Framework for*

Testbeds (DES-SERT) [15]. We use underlay routing (layer 2.5 routing, like MPLS [16]) of Ethernet frames in user space to send and receive packets. For upper layer protocols and the operating system kernel the gossip routing is fully transparent. We used *ARP-Requests* as packets sent with the `arping` command. The destination was set to an IP address that is not assigned for any mesh router in the network to achieve a one-way wave of packets. The packets are quite small even considering the overhead introduced by the underlay routing: 82 Bytes (ARP-Request, Ethernet, DES-SERT, and IEEE 802.11 MAC Frame). The source node sends 100 packets with a rate of $1/s$ for each value of the threshold $p \in (0..1]$. If the particular variant has no configurable parameter p as the probability is adaptive, the source sends 10,000 packets so that all experiments have the same duration. Each packet has a unique 16 bit sequence number that is used for loop detection. The packet rate is low enough to ensure that there is none or only limited media contention among packets with different sequence numbers while at the same time the experiment duration is kept within acceptable dimensions.

We selected node A and node B as source nodes. They are marked in Figure 1. Node A is a border node, while B is close to the center of the network. A has a node degree of 5 and B has 13 neighbors.

The parameter k, which is used by *gossip3* and *gossip6*, is set to 1. Higher values would lead to a flooding in significant parts of the testbed distorting the gossip routing as the network diameter is on average around ten hops. Considering the average node degree in the network, we deem an initial one-hop flooding high enough to ensure that the gossip routing does not terminate because only a limited number of neighbors around the source will (potentially) forward the packets. The value of the timeout parameter of *gossip3* (and *gossip6*) is not explicitly specified by the authors. We set it to 200 ms which is high enough for packets to arrive over alternative and potentially longer paths. At the same time the delay will only introduce limited contention among subsequent packets sent by the source. The value of T_{max} for the RAD timer of *gossip7* has been specified by the authors with 10 ms. In our first trials we noticed that this value was much too low in real world scenarios. This resulted in many nodes sending in a very short time frame and thus high redundancy. The parameter has to be tuned very carefully considering these circumstances to achieve low redundancy and low delay. As delay is not in the focus of this study, we used 100 ms which is an adequate trade-off in this scenario and should give an upper bound of the performance of *gossip7*.

The n_f parameter of *gossip9* was set to 14 by Abdulai. This value is based on the average node degree in a particular network given the parameters area of the network A, number of nodes N, transmission range R and using the following formula: $n_f = (N-1)\frac{\pi R^2}{A}$. In our case, the transmission range varies based on the locations of the nodes and the surroundings and the radio propagation is definitely not circular. From our experience we have an average (indoor) radio

Table 1. Parameters of the experiments in the DES-Testbed. Some parameters are only valid for particular variants. Common parameters are in the upper section, individual ones are in the lower section.

Parameter	Value(-s)	Gossip Variant(-s)
Number of Nodes	59	all
Topology	random, static nodes	
MAC Layer	IEEE 802.11	
Channel	13	
Source Nodes	A and B	
Threshold p	$p \in (0..1]$, in 0.01 steps	3, 6, 7
Packets per value of p	100, one per second	3, 6, 7
Packets	10,000, one per second	8, 9
Duplicate limit m	1	3, 6, 7
Timeout	200 ms	3, 6
Flooding for k hops	1	3, 6
T_{max}	100 ms	7
p_{min}, p_{max}	0.4, 0.9	8
n_f	4	9

range of about 11 m because of many thick walls which block the radio propagation. This does also consider in that we ignore long(-er) links with a low packet-delivery-ratio. Additionally it had to be compensated that the vertical radio propagation in between floors is worse than on the same level. As the routers of the testbed are not deployed in a plane, we calculate n_f as follows: $n_f = (N-1)\frac{\frac{4}{3}\pi R^3}{A \times h}$. With height $h = 13\ m$ and $A \approx 5600 m^2$ we get a result of about $n_f = 4$, which is of course only a rough estimate. The *gossip9* algorithm will fall back to flooding for all nodes with node degree ≤ 4.

Table 1 summarizes the experiment parameters. A full experiment run takes more than 28 hours including configuration overhead. Further experiments with different parameter settings will be considered as future work.

4 Evaluation

Please note that all graphs of this section and our first study are available on our website in higher resolutions[1].

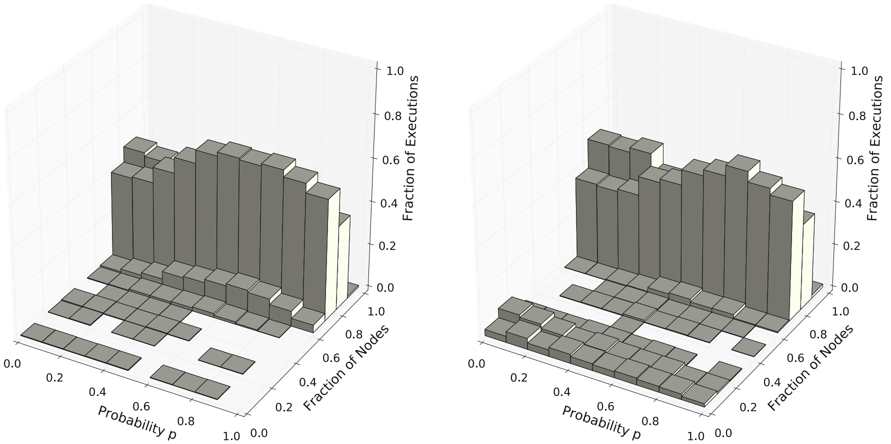
4.1 Figure Format Description and gossip3

For the evaluation, we consider in how many cases (executions[2]) when the source node sent a packet, which fraction of nodes did receive this packet. There are two

---

[1] Graphs: http://www.des-testbed.net/experiments/gossip

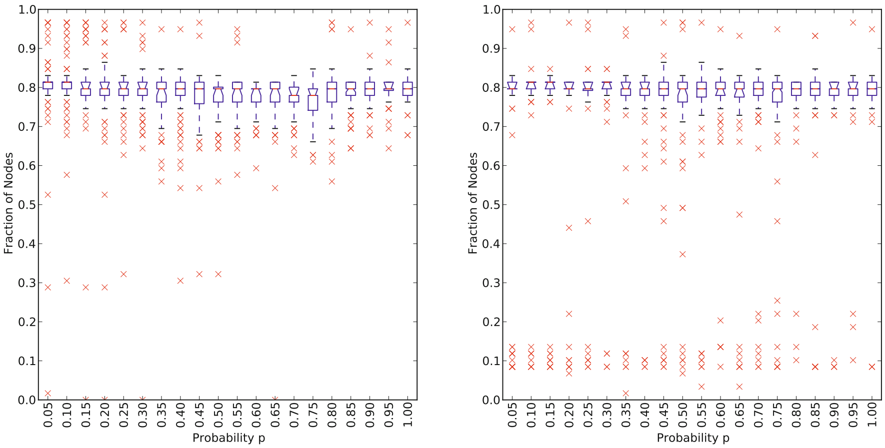
[2] The nomenclature of executions has been adopted from [6,7].



(a) Source Node A

(b) Source Node B

Fig. 2. Fraction of executions as a function of the fraction of nodes receiving the packets with *gossip3* for each value of p shown as histogram with bin size 0.1



(a) Source Node A

(b) Source Node B

Fig. 3. Fraction of nodes receiving the packets with *gossip3* as box-and-whisker-plot with median and 95% confidence interval

different representations of the data which will be explained based on the *gossip3* results. Figure 2 shows histograms for both source nodes with the parameter p on one of the axes. In Figure 3 the median fraction of nodes that received the packets is shown as a function of the parameter p. The plots show the median in a box-and-whisker plot with a 95% confidence interval; the whiskers extend up to 1.5 times of the inter-quartile range. All data points outside of the whiskers are considered statistical outliers. For *gossip3* we can see that the algorithm is independent of the configured threshold p and achieves a median of around 0.8. Further on as shown in Figure 2a, in more than 80% of the executions we reach a fraction of ≥ 0.75 of nodes. This means the algorithm in its current configuration performs very well especially considering that a router will not forward stored packets when one or more duplicates are received. Higher values of the parameter m might lead to an even larger median fraction of nodes. We consider a median of 0.8 to be the best that can be achieved in our testbed. From all experiments we learned that there are 11 nodes in the smaller building that are badly connected to the rest: this exactly matches the 0.8 fraction. Comparing *gossip3* from source node A and B we notice that for the latter there are more outliers which accumulate at 0.1 on the fraction of nodes axis (see Figure 3b). In these cases the 11 "problematic" nodes did not receive any packets from the source and no packet got to the larger building. Even the gossip routing variants discussed in the next section cannot resolve this problem but we conclude from all figures that this problem is independent from the value of p.

4.2 Advanced Gossip Routing

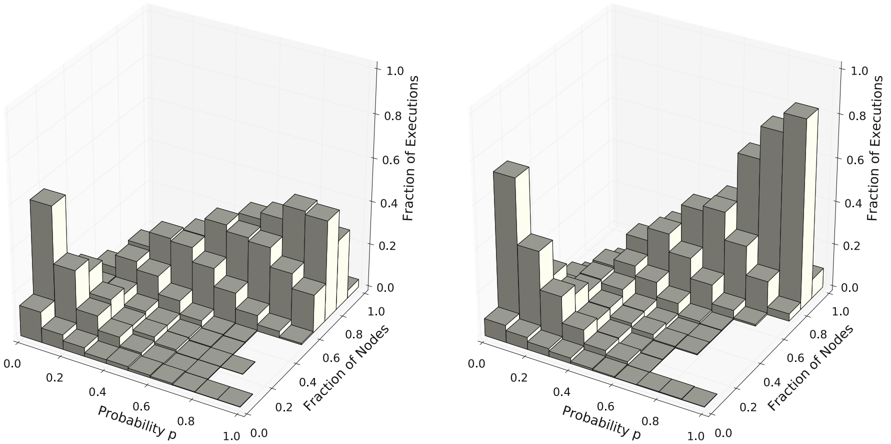
gossip6 from source A shows results (Figures 4a, 6a) similar to what we measured for *gossip1* in the last study[3]. The variant shows a strong dependence on the parameter p: the higher the value, the more nodes receive the packet from the source. Compared to *gossip1*, the whiskers are larger in *gossip6* especially for higher values p. This is probably an effect of the second broadcast chance that is missing in *gossip1*. From source B the median fraction of nodes makes a jump from below 0.2 to above 0.75 at $p = 0.75$. This is a bimodal behavior that we already observed in the first study. When comparing *gossip3* and *gossip6* we notice that, although the former is based on the latter, the results are totally different.

gossip7 shows results that are very similar to *gossip6* and considering the confidence intervals, the results are mostly equal. In Figure 6b we notice that for $p \geq 0.80$ the boxes get very small as well as the confidence intervals. The histogram for source A (Figure 4b) shows a much higher value for the 0.80 bin which underlines that the true median actually lies at 0.80. Overall the delayed forwarding combined with duplicate counting could only reduce the variance but not increase the reachability. Source node B shows similar differences in the histogram but no variance reduction in Figure 7b.

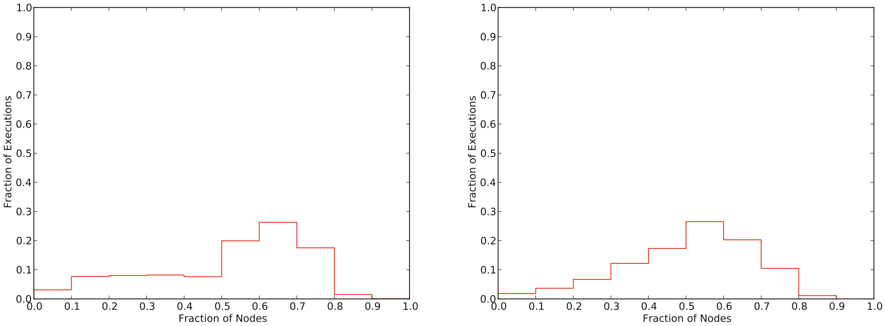
When the source is in the smaller building, neither *gossip3*, *gossip6*, nor *gossip7* were able to ensure that the gossip routing reaches the other building. Maybe the adaptive variants will perform better. As the last two gossip routing variants use no static value for p, their graphs show only a single histogram or box-and-whisker plot. The results are quite surprising. *gossip8* achieves only a median fraction of nodes of below 0.6 from source A and slightly above 0.1 from source B. Despite the large whiskers in Figure 6c and Figure 7c, it has to be

---

[3] *gossip1* uses only an initial k-hop flooding and forwarding based on p.



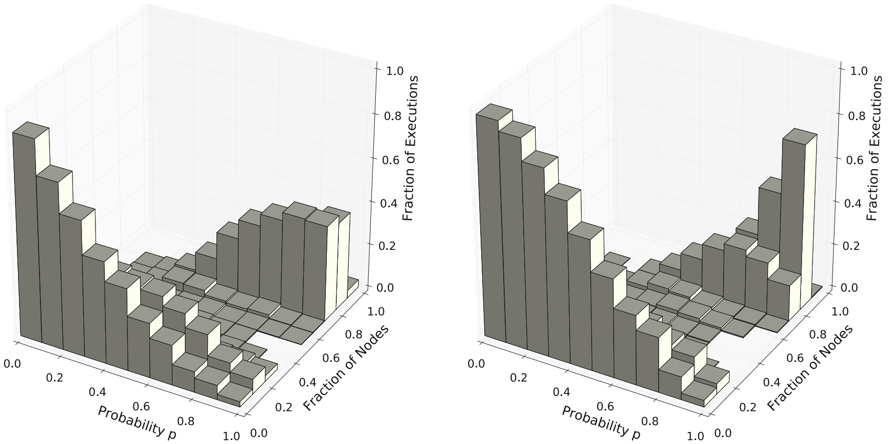
(a) *gossip6* (b) *gossip7*



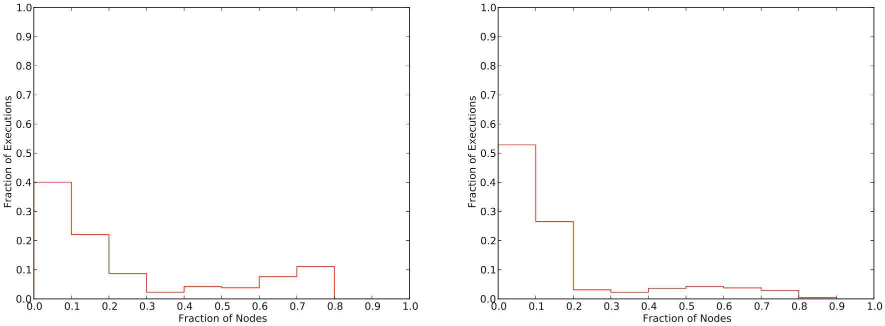
(c) *gossip8* (d) *gossip9*

Fig. 4. Source node A

assumed that the true median is in the 95% confidence interval. The confidence intervals are small as 10.000 data points are the basis of these graphs. Figure 4c shows that in 40% of the execution less than 50% received the packet from source A. *gossip8* has only one chance to forward the packets with the node degree n as the only factor that influences the probability. We learned that several nodes did not forward the packets as their node degree was too high. This is fatal for the gossiping as these nodes are required to connect the whole network - especially for source node B. The median fraction of nodes in Figure 7c with a value of about 0.1 corresponds to the well connected routers in the smaller building. Thus *gossip8* from source B did in many cases reach neither the 11 badly connected routers in the same building nor did the gossiping reach any router in the larger building. At least in our random network topology, the approach of *gossip8* has problems. *gossip9* seems to have the same problems as *gossip8*. The median fraction of nodes is actually worse for both sources (Figures 6d and 7d) and the histograms show a shift to the left (Figures 4d and 5d). The approach



(a) *gossip6* (b) *gossip7*



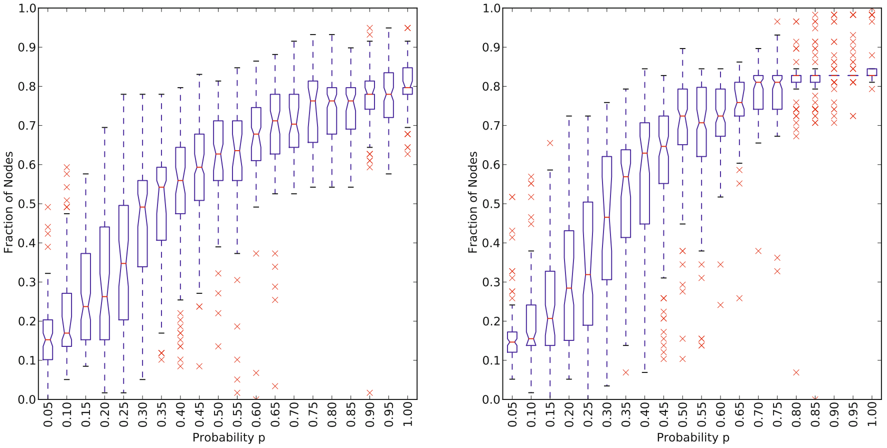
(c) *gossip8* (d) *gossip9*

Fig. 5. Source node B

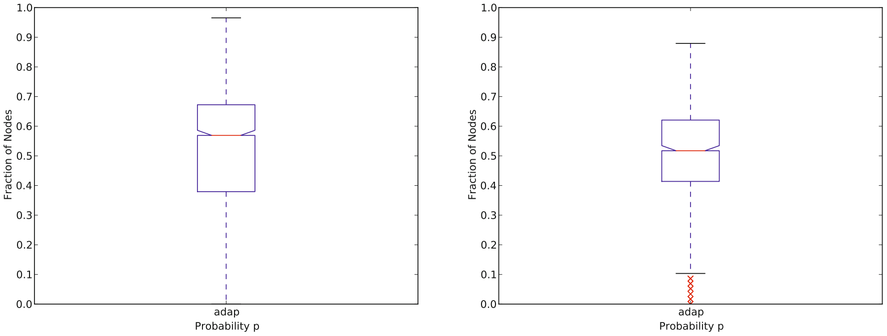
to consider the additional coverage that can be achieved by forwarding packets fails in our testbed as the distribution of nodes deviates from the assumptions made in the simulation study.

4.3 Redundancy

To evaluate the redundancy we plot the ratio of how many packets are sent when gossip routing is used to how many packets are sent with flooding: $pkts_{gossipX}/pkts_{flooding}$ Figure 8a shows the metric and its correlation with the reachability for source A and Figure 8b the same for source B. The intersection of the gray lines marks the data point for flooding which we measured in a separate experiment run. Therefore points in the top-left sector represent improvements. Variants with different probabilities p are shown as multiple, selected data points. *gossip3* and *gossip7* actually reach 80% of the nodes but with up to 30% fewer packets than flooding from source A. *gossip7* should be



(a) *gossip6* (b) *gossip7*

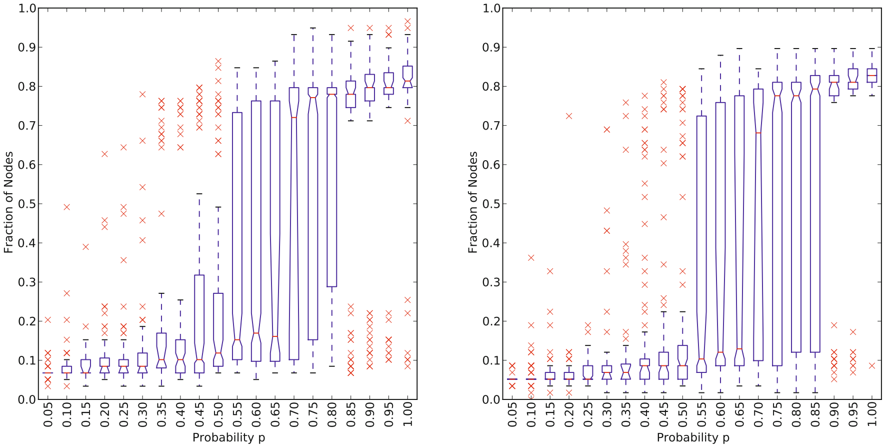


(c) *gossip8* (d) *gossip9*

Fig. 6. Source node A

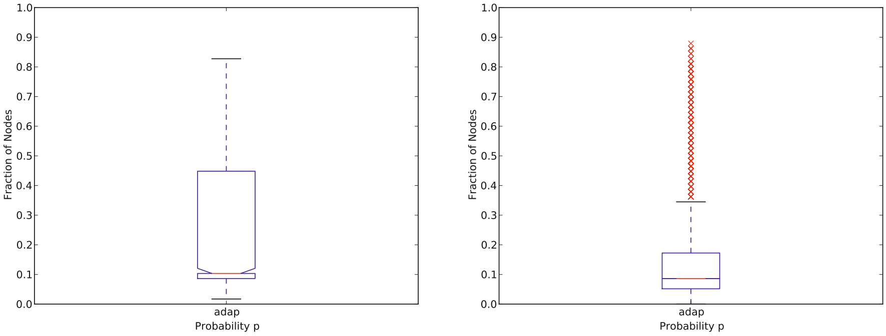
configured with $0.70 \leq p \leq 0.90$ and *gossip3* with $p \leq 0.90$, $m = 1$. For source B the savings are less (20%) but *gossip6* also shows better performance. *gossip6* and *gossip7* with $p = 0.9$ as well as *gossip3* with and $p \leq 0.90$, $m = 1$ show the best results. *gossip8* and *gossip9*, for which we also plotted the results of an additional experiment run, are in the bottom-left sector.

Duplicate-counting approaches seem to perform very well with some restrictions. *gossip6* decreases the number of forwarded packets in the source A scenario in comparison to *gossip3* on which it is based on, yet at the same time the reachability suffers. The influence of the parameter p is clearly visible in Figure 8a. *gossip6* is too conservative to determine the probability to forward stored packets. *gossip7*, which always stores packets and counts the duplicates, performs better and underlines that a static duplicate threshold m is good in our testbed scenario. In future experiments we will try to optimize the parameter T_{max} to achieve a lower delay while keeping the increased reachability.



(a) *gossip6*

(b) *gossip7*



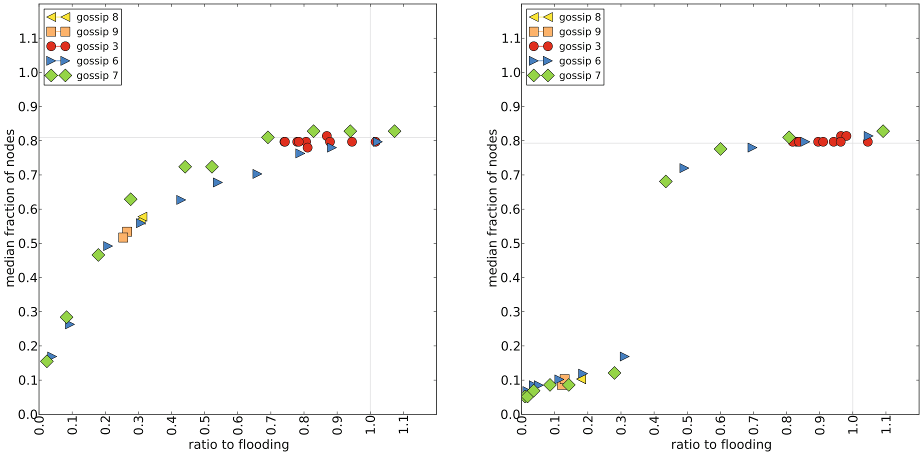
(c) *gossip8*

(d) *gossip9*

Fig. 7. Source node B

5 Conclusion

In this paper we evaluated four different gossip routing variants in the DES-Testbed. Despite the optimal experiment conditions (small packets, stationary nodes, single source, no other data flows, high timeout values) none of the discussed gossip variants could reach slightly more than 80% of the nodes in median. The number of packets could be reduced by up to 30% compared to flooding when the parameters are chosen adequately and the position of the source node is considered. Variants with adaptive approaches performed surprisingly bad as they seem to have problems in our topology with highly varying node degree. Duplicate-counting with a low threshold value ($m = 1$) performs very good and is totally overhead free as no management information is required, e.g., by using a HELLO protocol. The most important issue, identifying the nodes that are vital for the connectivity of the network, has yet to be considered to make gossip routing applicable in real world deployments. Gossip routing in wireless



(a) Source Node A (b) Source Node B

Fig. 8. Median fraction of nodes as a function of the number of forwarded packets in relation to flooding

networks requires further research to come up with an optimal solution that ensures high reachability and low redundancy independent of the source node position in the network.

References

1. Moy, J.: OSPF Version 2, RFC 2328 (Standard) (April 1998), `http://www.ietf.org/rfc/rfc2328.txt`
2. IEEE Std 802.11-2007. IEEE standard for information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks-specific requirements – Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, LAN/MAN Standards Committee, New York, NY, USA, pp. C1-1184 (June 2007), `http://dx.doi.org/10.1109/IEEESTD.2007.373646`
3. Ma, D.: Secure feedback service in wireless sensor networks. In: Dawson, E., Wong, D.S. (eds.) ISPEC 2007. LNCS, vol. 4464, pp. 116–128. Springer, Heidelberg (2007)
4. Vahdat, A., Becker, D.: Epidemic routing for partially-connected ad hoc networks, Duke University, Tech. Rep. CS-2000-06 (July 2000)
5. Blywis, B., Günes, M., Juraschek, F., Hofmann, S.: Gossip routing in wireless mesh networks. In: International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC 2010 (2010)
6. Haas, Z.J., Halpern, J.Y., Li, L.: Gossip-based ad hoc routing. In: Proceedings IEEE INFOCOM 2002, The 21st Annual Joint Conference of the IEEE Computer and Communications Societies, June 23-27. IEEE, New York (2002), `http://www.ieee-infocom.org/2002/papers/822.pdf`
7. Haas, Z.J., Halpern, J.Y., Li, L.: Gossip-based ad hoc routing. IEEE/ACM Trans. Netw. 14(3), 479–491 (2006)

8. Blywis, B., Günes, M., Juraschek, F., Schiller, J.: Trends, advances, and challenges in testbed-based wireless mesh network research. In: ACM/Springer Mobile Networks and Applications (MONET), special issue on Advances in Wireless Testbeds and Research Infrastructures (February 2010), `http://www.springerlink.com/content/hk25051135m36571/`
9. Shi, Z., Shen, H.: Adaptive gossip-based routing algorithm. In: Proceedings, 23rd IEEE International Performance, Computing, and Communications Conference IPCCC (2004)
10. Mohammed, A., Ould-Khaoua, M., Mackenzie, L.M., Perkins, C., Abdulai, J.-D.: Probabilistic counter-based route discovery for mobile ad hoc networks. In: IWCMC 2009: Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing, pp. 1335–1339. ACM, New York (2009)
11. Hanashi, A., Siddique, A., Awan, I., Woodward, M.: Performance evaluation of dynamic probabilistic broadcasting for flooding in mobile ad hoc networks. Simulation Modelling Practice and Theory 17(2), 364–375 (2009)
12. Abdulai, J.: Probabilistic Route Discovery for Wireless Mobile Ad Hoc Networks (MANETs) (2009)
13. Philipp, M.: A framework for distributed channel assignment in wireless mesh networks. Master Thesis, Freie Universität Berlin (March 2010)
14. De Couto, D.S.J., Aguayo, D., Bicket, J., Morris, R.: A high-throughput path metric for multi-hop wireless routing. In: MobiCom 2003: Proceedings of the 9th Annual International Conference on Mobile Computing and Networking, pp. 134–146. ACM, New York (2003)
15. Blywis, B., Günes, M., Juraschek, F., Schmidt, P., Kumar, P.: Des-sert: A framework for structured routing protocol implementation. In: IFIP Wireless Days 2009 (2009)
16. Rosen, E., Rekhter, Y.: BGP/MPLS IP Virtual Private Networks (VPNs), RFC 4364 (Proposed Standard) (February 2006), updated by RFCs 4577, 4684, `http://www.ietf.org/rfc/rfc4364.txt`