Telematics
Computer Systems

Freie Universität Berlin

Gossip Routing, Percolation, and Restart in Wireless Multi-Hop Networks

Bastian Blywis Philipp Reinecke Mesut Günes Katinka Wolter

Computer Systems and Telematics
Institute of Computer Science
Freie Universität Berlin, Germany
{blywis, preineck, guenes, wolter}@inf.fu-berlin.de

Technical Report
TR-NO: TR-B-11-08

October 17, 2011

Institute of Computer Science, Freie Universität Berlin, Germany

Contents

List of Figures

Abstract

Route and service discovery in wireless multi-hop networks applies flooding or gossip routing to disseminate and gather information. Since packets may get lost, retransmissions of lost packets are required. In many protocols the retransmission timeout is fixed in the protocol specification. In this technical report we demonstrate that optimization of the timeout is required in order to ensure proper functioning of flooding schemes. Based on an experimental study, we apply percolation theory and derive analytical models for computing the optimal restart timeout. To the best of our knowledge, this is the first comprehensive study of gossip routing, percolation, and restart in this context.

CHAPTER 1

Motivation

1.1 Introduction and Current State

Flooding is used by many network protocols to disseminate information, discover and advertise services, or to discover routes. Wireless multi-hop networks are inherently unreliable and packets can be lost. Furthermore, with gossip-based protocols, nodes will drop packets probabilistically [1]. Therefore, the process may fail to reach all nodes or a specific destination. Consequently, protocols employ retransmissions to ensure success, sending the same information again after a timeout: the process *restarts*[1].

This approach is applied by many reactive routing protocols like AODV [2] or DSR [3]. Other protocols send information periodically, i.e., the timeout happens continuously. For example, OLSR [4] floods *Topology Control* messages every 5 s by default so that all nodes have the required link state information. Although there is no success that can be signaled, e.g. by a reply, the timing is nevertheless crucial for the performance. As routing information is only stored for a certain holding time, packets should be received with a minimum rate or otherwise routes might not be available at particular times for some nodes.

In most protocols, the timeouts are static, pre-configured values that are defined in the protocol specification. These values are selected with a particular network size, diameter, topology as well as traffic and mobility patterns in mind, and do not adapt to changing environments. Furthermore, since there is often limited information about how the parameters were derived, manual adaptation is difficult.

Protocols either specify the deadline to complete the task or the maximum number of restarts. With an inappropriate timeout value protocol performance will be sub-optimal. These issues are not of theoretical nature but can be experienced in the real world, as we will show. For example, in an experimental study of DSR we observed many failed route discoveries [5] and thus high delays in higher level protocols. In these experiments and in this technical report we use the DES-Testbed, a multi-hop IEEE 802.11 indoor and outdoor network deployed on different floors of multiple buildings throughout our campus. The observed behavior was unexpected as the testbed topology is fairly densely meshed [6]. Furthermore, as we have shown in [7], the performance of flooding is dependent on the protocol parameters and also on uncontrollable environmental parameters as well as on load-dependent effects. In fact, in wireless multi-hop networks restarts are common and not rare events.

[1] Note that we use retransmission, restart, and retry synonymously.

In this technical report we study the effect of the restart timeout on an abstract flooding protocol that provides the same parameters as common protocols and allows us to focus on the flooding process. Existing protocols can be mapped to this abstract protocol, and optimizations obtained in its evaluation can be transferred back. We employ an empirical study and draw upon percolation theory and restart theory to evaluate the effect of the restart timeout and discuss analytical models for online optimization.

1.2 Structure

This publication is structured as follows. We provide the theoretical background to percolation and restart theory in Chapter 2. In Chapter 3, we study the effect of the restart timeout in a gossip routing protocol on the reachability. We give an outlook on our future work in online optimization in Chapter 4 and discuss open issues. Chapter 5 concludes the technical report.

CHAPTER 2

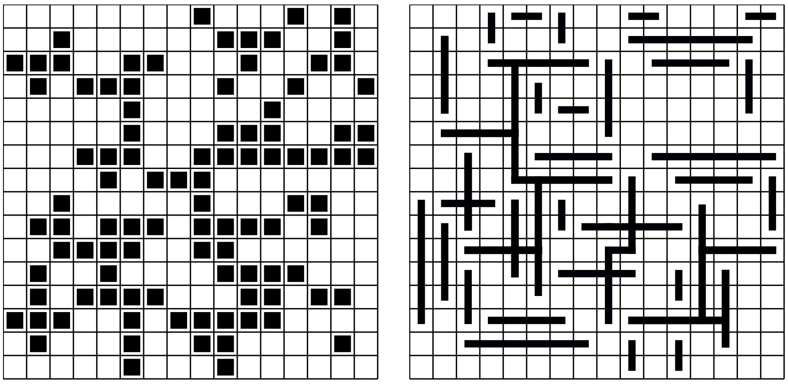
Related Work and Background

Gossip routing uses the gossiping model of spreading information: Nodes forward an incoming packet to their neighbors with a given forwarding probability p, or they drop the packet with $1 - p$ [8]. If the forwarding probability is 1, gossip routing is equivalent to flooding.

2.1 Gossip Routing and Percolation

Gossip routing is often referred to as a percolation problem [9]. Percolation theory focuses on the research of spatial random processes motivated by phenomenona in the real world. One example is percolation of water through porous material where cavities are independently and identically distributed with probability p. The central problem is to determine the critical probability p_c when percolation occurs, i.e., a path from the top to the bottom of the material exists with probability $\theta(p) = 1$. The system is either in a *sub-critical* or a *super-critical* state and experiences a *phase transition* at p_c. The critical value depends on the graph geometry as well as on the percolation model and has been determined analytically only for very few settings, e.g., the bond percolation in the square lattice [10]. Lattices are often studied where sites are either open with probability p or closed with probability $1-p$ and adjacent open sites create open clusters. This represents the so called *site model* that is depicted in the example in Figure 2.1a. In a second model, called *bond model*, bonds between sites of a lattice are either randomly open or closed as depicted in the example in Figure 2.1b. Both models can also be combined in the *bond-site model* [11] with two independent probabilities p_b and p_s. The random percolation process selects a subset of the bonds or sites from a graph.

Percolation theory may be applied to gossip routing by modeling nodes as sites and wireless links as bonds. However, so far no specific mapping of the random processes has been proposed. Nodes randomly drop or forward the packets, and open sites may represent nodes that have received the packet from a particular source. Packet loss may be modeled by the bond probability p_b. When the probabilities are larger than their critical values, percolation theory predicts that percolation occurs, i.e. high reachability can be achieved. The critical value derived from percolation theory may then be applied to reduce the number of forwarded packets, since for $p_c < p < 1.0$, reachability is maximized and the number of packets is reduced in comparison to flooding.



(a) Site model: Neighboring sites are connected if they are occupied

(b) Bond model: Neighboring sites are connected by bonds

Figure 2.1: Examples for the percolation process and two percolation models. The percolation process randomly selects (“occupies” or “creates”) sites or bonds.

2.2 Restart Theory

The restart method is a common approach to minimize the time until a particular task is successfully completed [12]. Applications include downloads, distributed queries, and TCP connections. Restart may reduce mean and variance of response times if the response-time distribution has high variance, but restarts occurring too often may result in overload.

Although the goal is similar to the application scenarios described in the introduction, there are some differences that require extension of the original approach. First, in the scenarios considered here there is typically no acknowledgment of task completion. Consequently, a response-time distribution cannot be obtained easily. Second, with restart one usually assumes independent and identically distributed (iid) trials. With changing parameters, interference, and mobility, this assumption is often not justified. Third, restart is built on the assumption that each new trial aborts the previous one, discarding all work previously performed. With routing protocols, however, a retransmission may reach new nodes, while the ones reached before will keep their data. This means that retransmissions are additive.

CHAPTER 3

Impact of the Restart Timeout

In the following we study the effect of the restart timeout on reachability under a fixed deadline T. We define reachability as the fraction of nodes reached for a given timeout τ, i.e.

$$R_\tau = \frac{\#reached}{\#nodes}$$

Other metrics, such as the time until the maximum number of destinations is reached or the required number of retries will be considered in future work.

The application behind our study is an abstract flooding protocol in the DES-Testbed [6], a wireless multi-hop network of 107 nodes spread over the campus of *Freie Universität Berlin.* We consider an undirected case without any signaling of success or failure, where packets are not addressed to a particular node and the process is continuous. By varying the forwarding probability p, the payload size B, the restart interval τ, the number of sources N, and the deadline T, we can use this scenario to model traffic characteristics of a wide range of routing protocols. All symbols that are used in this technical report are summarized in Table 3.1.

3.1 Experimental Evaluation

We start with an experimental evaluation of reachability without restart and with different restart intervals. To this end, we ran a set of experiments in the DES-Testbed, measuring reachability after a given deadline T.

First, we consider reachability without restart (R_∞) with deadline $T = 40$ s. Results for different configurations are shown in Figure 3.1. Note that reachability is lower than 1.0 for all configurations. It further decreases with decreasing forwarding probability p and increasing number of sources. While the former is directly related to the forwarding probability, the latter can be explained by collisions or increased noise levels due to the increased number of packets in the network [7].

In order to study the effect of the restart interval τ on reachability, we let each source emit the same packet every τ seconds. For deadline $T = 15$ s we set $N = 50$ and increase τ from 0.05 to 5 s. For improved legibility, we split the results into the two plots in Figure 3.2. The left-hand plot shows reachability for $\tau \in [0.05, 0.2]$ with step-size 0.025 s, while the right-hand plot gives measurements for $\tau \in [0.02, 5]$ with a coarser step-size of 0.2 s. Observe that reachability grows as τ increases until it reaches a plateau at around $\tau = 0.2$ s. For all forwarding probabilities, the reachability starts to decrease again at

Symbol	Description
B	Payload size
$E[S]$	Expected network sojourn time of a packet
f	Suppression factor
k	Parameter of the binomial distribution
l	Number of successes / arriving packets
m	Load threshold
N	Number of simultaneous sources
p	Forwarding probability
p_c	Critical probability
q_d	Loss probability independent of τ
q_e	Probability exactly one packet arriving
q_o	Loss probability due to overload
q_r	Reachability without restart
q_s	Probability of at least one packet arriving
T	Deadline
R	Reachability
τ	Restart interval

Table 3.1: List of the used mathematical symbols

some τ, with higher forwarding probabilities having a longer plateau. The increase and decrease can be explained as follows: For small τ, each node emits a large number of messages, thereby causing collisions and an increased noise level throughout the network. Both factors result in a decreased probability of successful message transmission [7]. On the other hand, large τ translate to a small number of retries before the deadline, resulting in lower reachability. Both effects will be explored in more formal terms in the following sections. Note that the forwarding probability has different effects on the left-hand and right-hand sides of the reachability curve. For small timeouts, lower forwarding probability translates to a constant offset, while with large timeouts smaller forwarding probabilities give a lower drop-off point and steeper decrease in reachability.

Summarizing the experimental results, we can state that there exists a range of restart values τ that maximize reachability. Restarting too late does not improve reachability, while restarting too early reduces it due to increased interference.

3.2 Percolation Theory

Considering reachability in the testbed as a percolation problem, Figure 3.1 illustrates that the network does not percolate and that there is no phase transition This can be explained by the finite size and the unreliability of the network.

To exclude collisions and interferences as possible sources of this effect, we run graph-based gossip routing simulations using the testbed's topology both with and without lossy and directed links based on the data that was measured in a previous study [6]. Graph-based simulation means that interference on the medium are not modeled. The probability to successfully transmit a packet over an edge, i.e., link between two nodes is determined by the packet delivery ratio that was achieved for this pair of nodes in the testbed. Packet collisions are not possible as each packet is evaluated independently. In

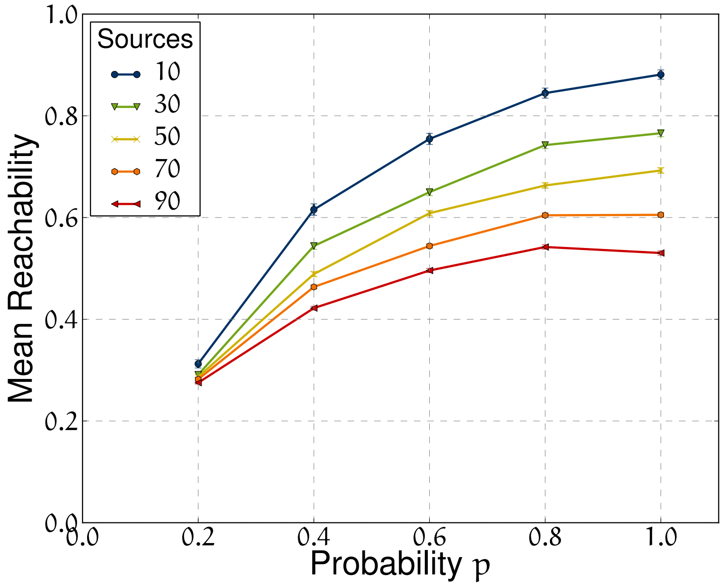


Figure 3.1: Reachability without restart in the testbed with different forwarding probability p and different numbers of simultaneous sources N (payload size $B = 0$ bytes).

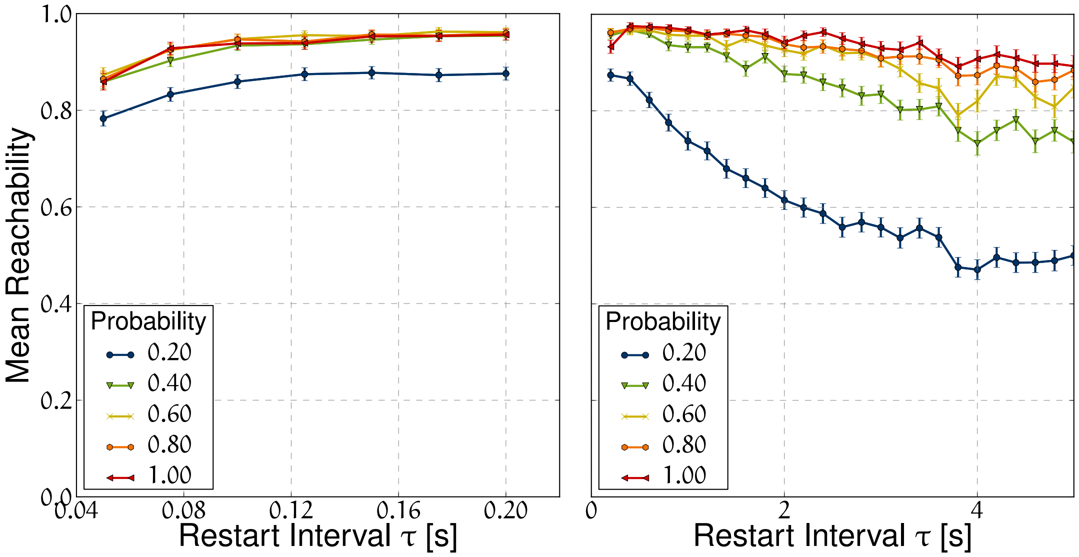


Figure 3.2: Reachability in the testbed after deadline $T = 15$ s with restart interval τ ($N = 50$, payload size $B = 0$). Each graph shows the reachability that was achieved with a particular forwarding probability p.

the simulation each node emitted 100 packets that were forwarded by the gossip routing protocol.

As shown in Figure 3.4, there is also no (rapid) phase transition but the more the particular scenario abstracts from specific properties of the topology, i.e., unidirectionality or packet losses, the more the results match the expectations based on the percolation models. Nevertheless, the graph for the directed, lossy topology that describes the mean reachability achieved from all nodes matches the results from the testbed that can be achieved when 10 (or less) sources send simultaneously but the data does not match with the results of larger numbers of sources. This corroborates the measured results and illustrates that interference and collisions have to be included in the percolation model to

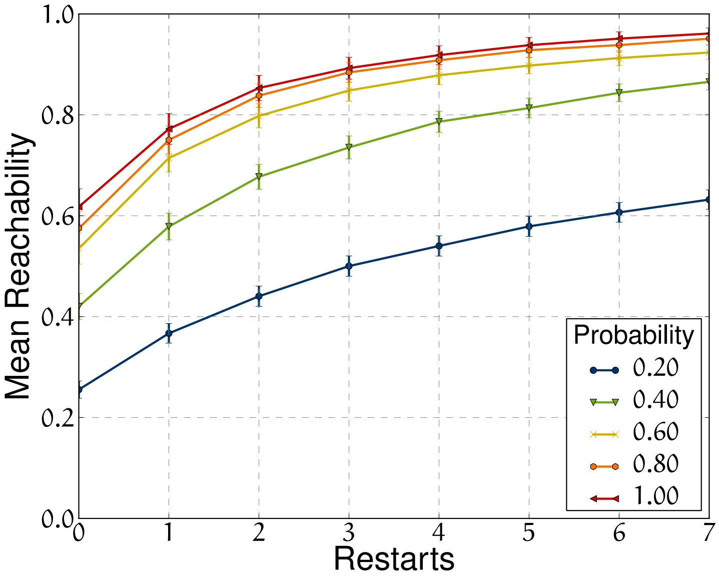


Figure 3.3: Reachability in the testbed after k restarts ($N = 50$, $\tau = 5$ s, and $B = 0$ bytes).

derive accurate predictions.

We propose a suppressed bond-site model [13] where the site probability p_s suppresses, i.e., decreases the bond probability p_b. The forwarding probability p represents the site percolation probability p_s in this model. The probability p_b for open bonds models successful transmissions and decreases when the number of open sites, i.e. transmitting nodes increases above a particular threshold m by a factor given by function f

$$\begin{aligned} f(p_s) &= \begin{cases} 1, & \text{if } \nu_t(p_s) < m \\ c(\nu_t(p_s)), & \text{else} \end{cases} \\ c(\nu_t(p_s)) &\to [0,1), \end{aligned}$$

where $\nu_t(p_s)$ is the mean number of transmitting nodes in the interference range when forwarding probability $p = p_s$ is used and $c(\nu_t(p_s))$ represents the probability of a packet loss due to collisions or increased noise level. Thus the lower reachability for small τ (higher load) as shown in Figure 3.2 can be modeled by the proposed model. As we see in Chapter 4, the number of transmitting nodes is actually a too abstract metric when other parameters change. This model will be further studied in future work.

Yet there remains one particular difference between gossip routing and percolation models. Although the system is in the sub-critical state, we nevertheless can achieve high reachability when restarts are involved, as shown in Figure 3.3. Each succeeding packet may be received by a different set of nodes. This means that with each trial different (connected) subgraphs containing the source are selected, whose union eventually becomes a subgraph that contains the same vertices as the original graph. This phenomenon is not considered in percolation theory. Note that Figure 3.3 illustrates that the reachability gain decreases with each restart. We expect to find a power-law relationship, but this is left as future work.

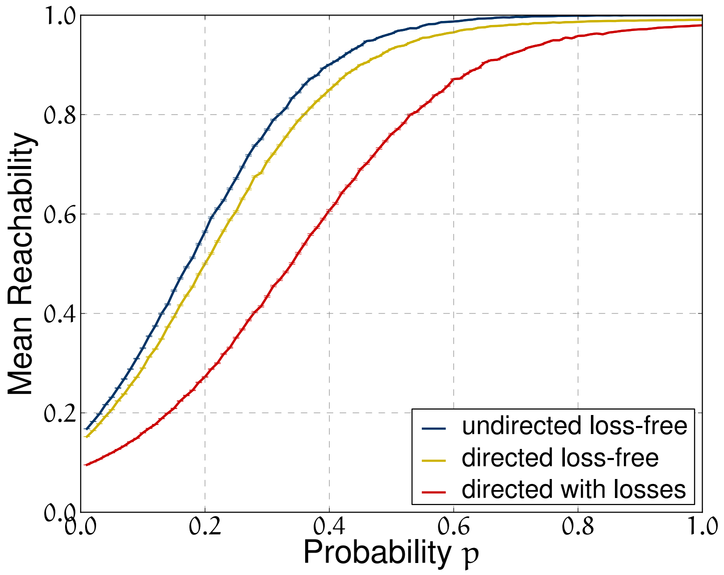


Figure 3.4: Graph-based simulation of gossip routing using the DES-Testbed's topology.

3.3 Restart Theory

We now derive an abstract analytical model for the effect of the restart timeout τ on the reachability R_τ. The model is based on the behaviour of the system, as seen by an arbitrary node sending packets.

First, note that each packet transmission has a certain success probability q_r and thus can be considered a Bernoulli trial. For a single trial, q_r is equal to the reachability without restart, R_∞. Our measurements demonstrated that the probability of successful packet transmission depends on the network load, which in turn depends on τ. Furthermore, without retransmissions the probability of successful transmission is lower than 1. Let $q_o(\tau)$ be the probability of loss due to overload or interference when the restart interval is equal to τ. Let q_d be the probability of loss due to influences independent of τ. Then, the success probability is

$$q_r(\tau) = (1 - q_o(\tau))(1 - q_d).$$

Second, restart with restart interval τ and deadline T implies $k = \lceil T/\tau \rceil$ Bernoulli trials, each with success probability q_r. Assuming constant q_d in the interval up to T, the probability of l successes (i.e. l messages arriving) follows a Binomial distribution with parameters k and q_r and probability mass function

$$f(l, k, q_r(\tau)) = \binom{k}{l} q_r(\tau)^l (1 - q_r(\tau))^{k-l},$$

and thus the probability of *at least* one message arriving is

$$\begin{aligned} q_s(\tau) &= 1 - f(0, k, q_r(\tau)) = 1 - (1 - q_r(\tau))^k \\ &= 1 - f(0, T/\tau, q_r(\tau)) = 1 - (1 - q_r(\tau))^{T/\tau}. \end{aligned}$$

Note that $q_s(\tau)$ expresses reachability with restart interval τ, i.e. $R_\tau = q_s(\tau)$. We might also be interested in the probability of *exactly* one message arriving, which is

$$q_e(\tau) = f(1, T/\tau, q_r(\tau)) = T/\tau q_r(\tau)(1 - q_r(\tau))^{T/\tau - 1}.$$

Using an upper-bound approximation, the same approach may be followed if we permit slight variations in q_d in the interval up to the deadline T. However, it requires heavier notation and is omitted here.

In order to obtain the probability of no loss due to overload, $1-q_o(\tau)$, we approximate the network as an $M|G|\infty$ queue, i.e., we assume that the arrival process is Poisson, network sojourn times can be described by a general distribution with expectation $E[S]$, and all arriving jobs (messages) are served (transmitted) in parallel (cf. [14]). We model overload by a threshold $m > 0$ for the queue length. If there are at least m jobs in the queue, jobs will still be served (i.e. the queue length is not bounded and the queue does not change its behavior), but service is faulty, rendering the completed jobs useless. This models the overload situation where messages are still being transmitted, but rendered unusable by collisions and noise, as described above. Then probability $p_i(\tau)$ of a queue length i given a restart interval τ is [14]

$$p_i(\tau) = \frac{e^{-\rho(\tau)}}{i!}\rho(\tau)^i,$$

where $\rho(\tau) = \lambda(\tau)E[S]$ is the load of the queue, and $\lambda(\tau) = {}^N\!/\!_\tau$ is the rate of the arrival process for N nodes performing restart. Consequently, the probability of no overload is

$$(1 - q_o(\tau)) = \sum_{i=0}^{m-1} p_i(\tau).$$

Figure 3.5 shows reachability derived from the analytical model, using parameters close to those observed in the testbed. The predicted behavior for q_s (at least one packet arrives) closely matches that observed in the measurements (Figure 3.2): For small timeout values τ, reachability is low, as τ grows it reaches a plateau, before dropping again. Note, however, that the decrease for low τ is much more drastic than in the measurements. This is due to two pessimistic assumptions: First, in the model all simultaneous messages contribute to overload. This is not the case in the real network, where messages only cause overload if they are sent within the same interference range. Second, the model assumes that for load larger than the threshold all messages are lost. In the real network some messages might still be submitted successfully, and, furthermore, message loss due to overload might itself alleviate the overload situation.

Figure 3.5 also shows the probability of exactly one packet arriving. Although q_e is typically not relevant in a real network, this curve provides an engineering rule for selecting a timeout that maximizes reachability while avoiding network load in cases where there is a wide plateau in the curve for q_s (i.e. a large range of optimal τ): As τ grows, $q_e(\tau)$ reaches a first maximum, then drops, before reaching a second maximum and dropping again. The valley between both maxima corresponds to the case where more than one packet arrives. Since for timeout values to the left of this valley there are more packets transmitted, but still only one arrives, the maximum to the left implies loss due to overload. Consequently, the right-hand maximum (larger timeout values) should be chosen. When optimizing q_s, one should thus choose a timeout value from the right-hand side of the interval maximizing reachability.

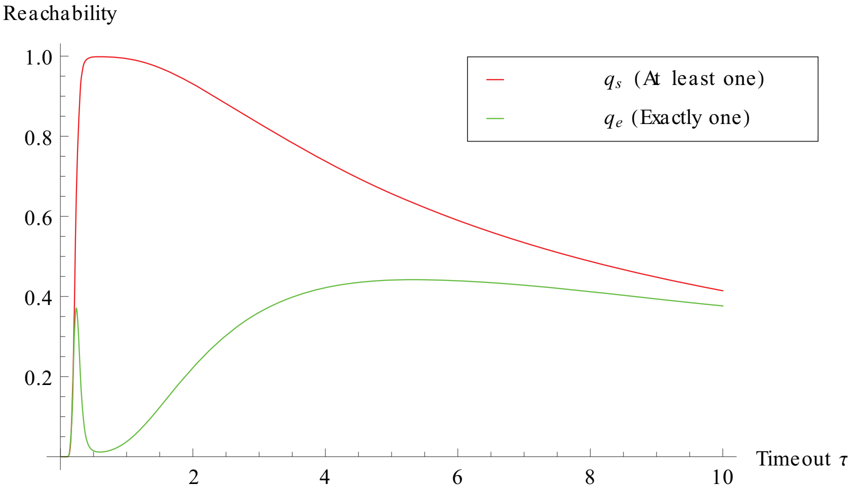


Figure 3.5: Reachability according to the analytical model ($N = 50, q_d = 0.7, E[S] = 0.05$ s$, m = 5, T = 15$ s).

CHAPTER 4

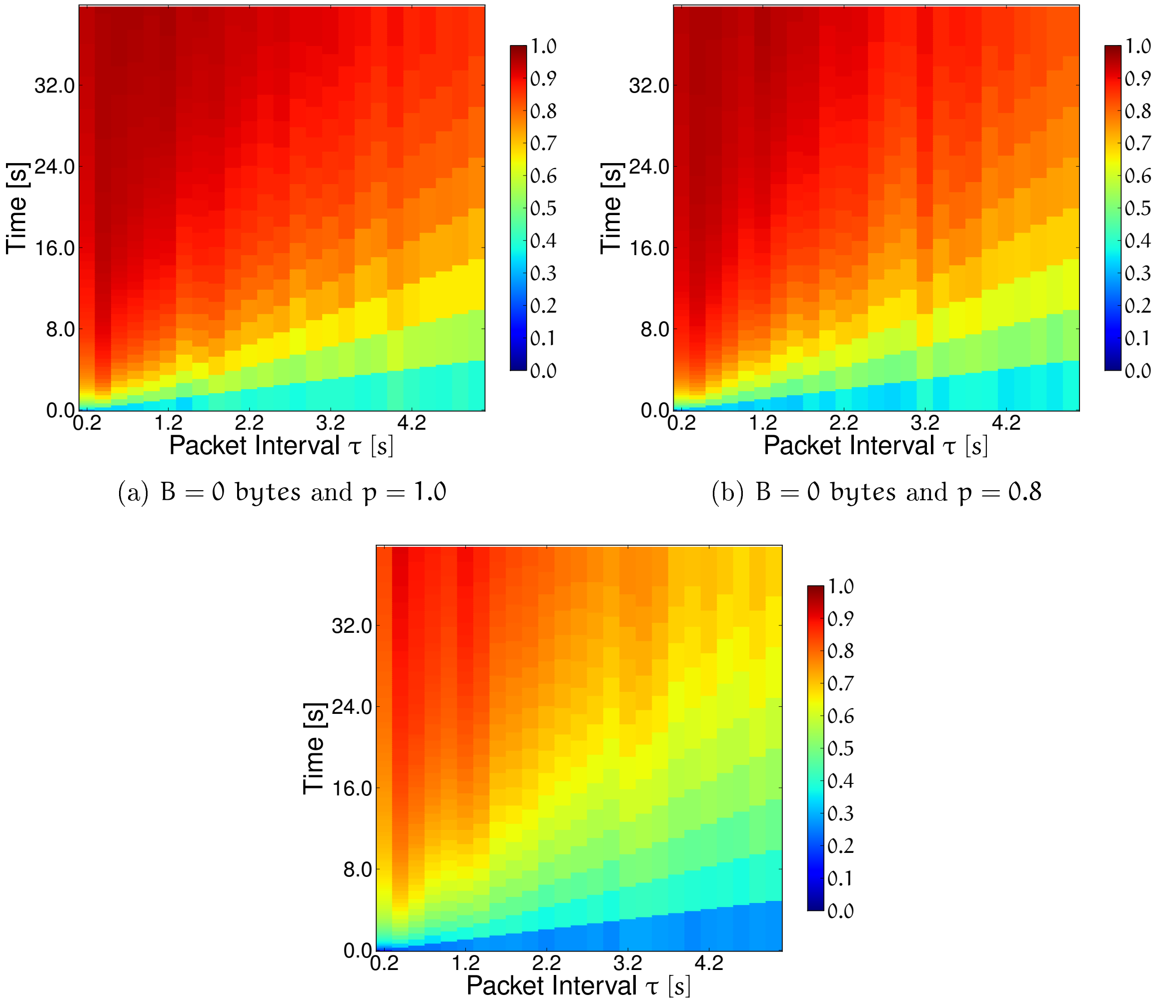
Future Work

4.1 Algorithms for Timeout Optimization

In the previous sections we observed that reachability depends on the restart timeout τ, and that there exists a range of values for τ for which reachability is maximal. As illustrated in Figure 4.1 and Figure 4.2, in practice the optimal timeout depends on a wide range of parameters, such as the deadline of the protocol, the forwarding probability, the number of sources and the payload size. Note that the graph for $p = 0.8$ in Figure 3.3 actually is a vertical slice through Figure 4.2, while the graph for $p = 0.8$ in Figure 3.2 is a horizontal slice through Figure 4.2, i.e., for a specific deadline that is application dependent.

Our current work focuses on developing distributed optimization algorithms that provide an optimal timeout τ at run-time, based on observations of the network. One such algorithm follows from the analytical model presented in Section 3.3, which allows straightforward computation of the optimal timeout once the parameters are known. Then, the main problem is the estimation of these parameters at run-time as distributed network monitoring is a major challenge. Similarly, the application of the suppressed bond-site percolation model also requires better estimation of network parameters.

The central problem is that the network state can only be estimated based on asynchronous and unreliable communication between the nodes and that a node cannot easily differentiate between a node that has left the network, e.g., it was switched of and between a node that cannot be reached anymore due to an overload situation. Presumably, specific invariants have to be assumed, e.g., that the run-time algorithm will always keep the network from overloading so that nodes that join of leave the network can either reliably estimate the network state or advertise their intentional leaving. Nevertheless, the estimation of the network state will have to rely on observations over a particular observation period and advertisement of the gathered information which introduces additional problems. For example, the period has to be brief enough that the network state can be considered stable during the observation time yet a sufficiently large number of received (and sent) packets are required for fine-granular estimations. As the rate of advertisements, probe packets, normal data packets cannot be increased infinitely due to the possible overloading of the network, only limited and partial information might be available.



(a) $B = 0$ bytes and $p = 1.0$

(b) $B = 0$ bytes and $p = 0.8$

(c) $B = 128$ bytes and $p = 0.8$

Figure 4.1: Pseudo-color plot of reachability in the testbed for deadlines $T \leq 40s$ on the time axis and restart interval τ for different payloads B and forwarding probabilities p. High reachability is represented by red color and low by blue. $(N = 90)$

4.2 Model Extension

The restart theory based model discussed in Section 3.3 can be used to model the reachability that is achieved with gossip routing but one particular difference remains. The expected reachability is shown as a function of the network load in Figure 4.3. The solid blue graph represents the expectation based on the model whereas the dashed red graph shows the achieved reachability in a real-world network. The area between the two graphs represents this modeling error.

For particular configurations of the gossip routing protocol, i.e., the forwarding probability, deadline, and payload size the network will be overloaded. In this state packet collisions and interference become significant so that the communication is disturbed and the reachability decreases. In the restart theory based model that is based on the $M|G|\infty$ queue, all jobs are faulty when the load exceeds the threshold m and the reachability immediately drops to zero. Of course, the overload state does not induce a phase transition in wireless multi-hop networks. As depicted in Figure 4.3, the decrease in reachability in-

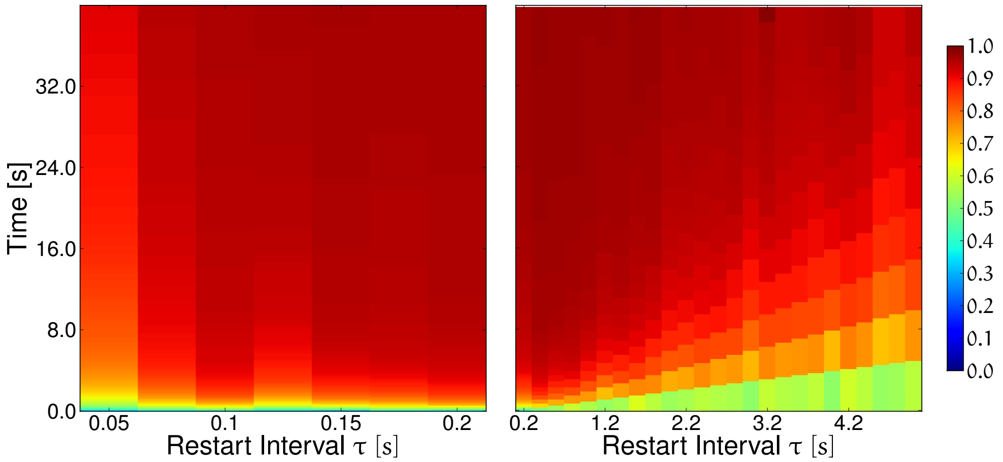


Figure 4.2: Pseudo-color plot of reachability in the testbed for deadlines $T \leq 40s$ and restart intervals τ. Red implies high reachability. $(N = 50, B = 0, p = 0.8)$

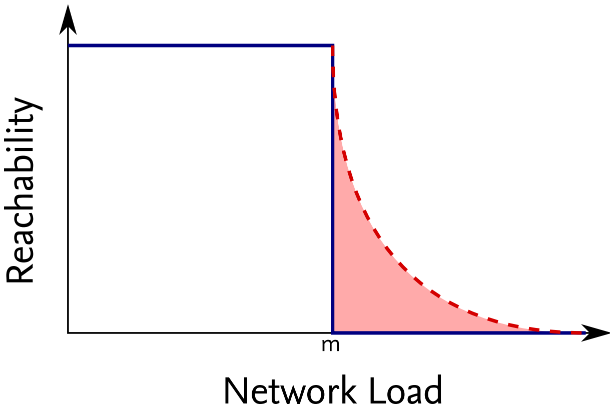


Figure 4.3: Deviation of the models when the network becomes overloaded at threshold m. The solid blue graph represents the expectation based on the model whereas the dashed red graph shows the achieved reachability of the gossip routing.

creases the more the network load increases beyond the threshold. This can be explained by the topology of the network. Depending on the topology, packets may not reach specific areas that experience higher load than the others. For example, this may be an effect of a higher average node degree in the area and therefore increased interference and increased probability for packet collisions. Alternatively, if the load is uniform in the network, packet transmission between the source at its neighbors may still be possible to some extent but the forwarding to the 2-hop neighbors may fail as each hop increases the chance that a packet gets lost. In this scenario the load acts as a negative amplification factor on the link PDR, similar to the suppression factor f as discussed for the suppressed percolation model in Section 2.1.

Thus, the model has to be extended to provide a parameter to configure respectively model the slope of the reachability in the overloaded network. The monitoring of the network state and the derivation of the required parameter(-s) is a non-trivial task as discussed in the previous section.

CHAPTER 5

Conclusion

In this technical report we studied information distribution in wireless multi-hop networks based on flooding and gossip routings. As we have confirmed in our empirical study, retransmissions are required to maximize reachability. We have shown that reachability with a deadline can be maximized by choice of an optimal restart timeout. The negative effect of too high or low restart timeouts was measured in experiments. We proposed a percolation model to include the effects of the shared medium and an analytical restart model for predicting reachability for a given timeout. Finally, we sketched a method for distributed on-line optimization of reachability by choice of the timeout and discussed open problems. Gossip routing, percolation, and restart were studied in a novel way based on experiments in a testbed, graph-based simulation, and an analytical approach.

Bibliography

[1] R. Bakhshi, "Gossiping models," Ph.D. dissertation, Vrije Unversiteit Amsterdam, 2011.

[2] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," RFC 3561 (Experimental), Jul. 2003.

[3] D. Johnson, Y. Hu, and D. Maltz, "The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4," RFC 4728 (Experimental), Feb. 2007.

[4] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," RFC 3626 (Experimental), Oct. 2003. [Online].

[5] B. Blywis, M. Günes, D. J. H. Gutzmann, and F. Juraschek, "A testbed-based study of uni- and multi-path dynamic source routing in a wmn," in *Wireless Days (WD), 2010 IFIP*, oct. 2010, pp. 1 –5.

[6] B. Blywis, M. Günes, F. Juraschek, O. Hahm, and N. Schmittberger, "Properties and Topology of the DES-Testbed (2nd Extended Revision)," Freie Universität Berlin, Tech. Rep. TR-B-11-04, July 2011.

[7] B. Blywis, M. Güneş, F. Juraschek, and O. Hahm, "Challenges and limits of flooding and gossip routing based route discovery schemes," in *The 36th IEEE Conference on Local Computer Networks (LCN)*, 2011.

[8] Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip-based ad hoc routing," *IEEE/ACM Trans. Netw.*, vol. 14, no. 3, pp. 479–491, 2006.

[9] G. Grimmett, "Percolation and disordered systems," in *in Lectures in Probability Theory and Statistics, Ecole d'Ete de Probabilites de Saint-Flour XXVI-1996.* Springer, 1996, pp. 153–300.

[10] T. E. Harris, "A lower bound for the critical probability in a certain percolation process," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 56, pp. 13–20, 1960.

[11] M. Yanuka and R. Englman, "Bond-site percolation: empirical representation of critical probabilities," *Journal of Physics A: Mathematical and General*, vol. 23, no. 7, p. L339, 1990.

[12] K. Wolter, *Stochastic Models for Fault Tolerance - Restart, Rejuvenation and Checkpointing.* Springer Verlag, 2010.

[13] L. Nduwayo, R. Lindebaum, and N. Chetty, “Suppressed bond-site percolation,” *Computer Physics Communications*, vol. 180, no. 4, pp. 503 – 508, 2009, special issue based on the Conference on Computational Physics 2008 - CCP 2008.

[14] R. Jain, *The Art of Computer Systems Performance Analysis : Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley, New York :, 1991.