How to Model and Analyze Gossiping Protocols?

Joost-Pieter Katoen
Software Modeling and Verification Group, RWTH Aachen University, Germany

1. INTRODUCTION

During coffee breaks, birthday parties, and even sometimes at business meetings, gossiping is a common acitivity in which most people kindly engage. Gossiping forms one of the oldest and most common means of spreading and sharing facts and views. It has, however, a somewhat negative connotation as most people understand gossip as a means to spread dust and misinformation, as, e.g., a discussion of scandals of (mostly female) Hollywood celibrities (see, e.g., `topgossips.com`). In this special issue of ACM PER, however, gossiping is interpreted in a more positive sense, namely, to spread information such as measured data, routing information or the like, in distributed systems. The main focus is on modeling and quantitative analysis of such gossiping protocols.

What is a gossiping protocol? Put in a nutshell, in a gossiping protocol, nodes in a distributed system disseminate information in a way that mimics how a group of people spreads gossips. A node continuously exchanges information about the nodes they know about (called their "view"). On receipt of information of other nodes, a node updates its local view according to some recipe, e.g., by replacing old information by more recent one, or by some way of merging the already available and newly available information. (This very much resembles the way people deal with gossips.) A scheme in which a node both sends and receives information is called a push-pull policy; other variants are simple push, or pull policies.

Quantitative properties of gossiping protocols? Let's explain the kind of properties one wants to establish of gossiping protocols by means of a simple example—the gossiping girls. This problem has originally been raised at the Dutch Science Quiz, an annual, TV-broadcasted event. There are n girls. All know one gossip which is unknown to all others. Each girl is extremely desperate to disseminate her gossips and (how human can one be?) to hear the other gossips. The girls communicate via a two-way channel, a handy say. When two girls phone, they exchange all their gossips. What is the minimum number of phone calls needed so that every girl knows all gossips? Such questions are typical for gossiping protocols: when is all information spread to all nodes in the network? Can this be optimized? What is the minimum distance between a pair of nodes? And so forth. A simple scheme for the gossiping girls example that takes $2n-3$ phone calls is that each girl phones one specific girl, Alice, say, to inform her about the gossips. Alice subsequently phones every girl except the last one that phoned her. But, for some cases there is a scheme that requires one phone call less! [1]

What's the challenge? The main difficulty of gossiping protocols is their size —the number of nodes can be many thousands or more— the large number of parameters in the protocol and the fact that these protocols canot be analyzed by considering individual node behaviour only, i.e., a lack of compositionality. Let us discuss the number of parameters in some more detail. There are, e.g., different update policies to amend local views on receipt of new information, the view size is of importance, i.e., how much information can a node keep track of, how to select the next node to inform it about my information, and so on. More importantly, though, the networks are typically highly *dynamic*, i.e., nodes enter and leave the network spontaneously, either due to failure or in case of a wireless network, nodes may get out of reach. This complicates matters significantly. The sketched solution to the gossiping girls exercise is for instance not very robust: once Alice becomes unreachable a different scheme has to be realized. Due to these complicating characteristics, gossiping protocols are typically analysed by means of emulation or simulation. Focus of this special issue is to apply numerical and/or analytical techniques to assess quantitative properties of gossiping protocols.

[1] One can prove that at least $2n-4$ phone calls are needed. The proof of this fact is not immediate; only two correct solutions were sent in after posing this question in the Dutch Science Quiz in 1999.

2. MODELING

Explicit modeling of node behaviour. The papers by Crouzen *et al.* [4] and Kwiatkowska *et al.* [9] both study a gossiping protocol by Jelasity *et al.* [7] and take a rather similar modeling approach: each node is modeled by a probabilistic finite-state automaton (a Markov chain, in fact), and the entire system is modeled by a parallel composition of these automata, synchronizing corresponding sends and receipts. Whereas [9] uses the concept of interleaving for the parallel composition, [4] exploits randomization by incorporating a random delay prior to the node's dissemination of information. The former approach yields a Markov decision process (MDP) that, due to the presence of nondeterminism, is subject to worst case and best case analysis. That is to say, we obtain minimal and maximal probabilities under all possible resolutions of the scheduling of the individual nodes—when is which node performing a step? In addition, the schedule that yields these extreme probabilities is obtained. That is to say, the optimal schedule can be synthesized. The latter approach using randomized composition yields a continous-time Markov chain (CTMC). Results in this approach can be interpreted as average results as the resolution of the nondeterminism is done probabilistically. The distinct modeling of concurrency is one of the main differences between [9] and [4].

How to obtain state-based models? The probabilistic automata are obtained from higher-level formalisms. Crouzen *et al.* [4] take two formalisms: process algebra and graph grammars. The advantage of the former is the possibility of symbolic manipulation, whereas graph grammars allow for symmetry reduction (e.g., one abstracts from node identities) and provide a rather natural means to describe the dynamic network behaviour. Unfortunately, the probabilistic information is not directly supported and needs to be incorporated at a later stage in the analysis. The relation between the process algebra and graph-grammar model is also not studied which makes it hard to compare the approaches. Kwiatkowska *et al.* [9] provide a more detailed model and use a probabilistic variant of reactive modules, a state-based description language. For n=4 (number of nodes) and local view size c=2, they obtain a state space which is three orders of magnitude larger than [4].

The main drawback of these two approaches is that the behaviour of each individual node is explicitly given, yielding enormous state spaces even for simple instances, for just half a dozen nodes, say, and a static network topology. Modeling the dynamic behaviour of the network will yield even larger state spaces. It seems that abstraction techniques such as partial-order reduction or more aggressive, semi-automated abstraction techniques are indispensable to enable the modeling of larger system instances.

Matrix modeling. The other two papers in this volume take a radically different modeling approach. Krieger *et al.* [8] use matrices to model the "state" of the entire distributed system. This novel approach can be best illustrated by the simple gossiping girls example. The main idea is to model the state of the entire system by a boolean matrix $\mathbf{G}$, say, such that $\mathbf{G}(i, j) = 1$ if and only if girl i knows the gossip of girl j. For three girls, we then obtain the state space as indicated in Figure 1, where a state is a matrix and self-loops indicate a phone call in which no new gossips are spread. This happens, e.g., when two girls phone each other but find out that they are aware of the same gossips. This simple approach as put forward in [10] is now adapted in [8] to deal with:

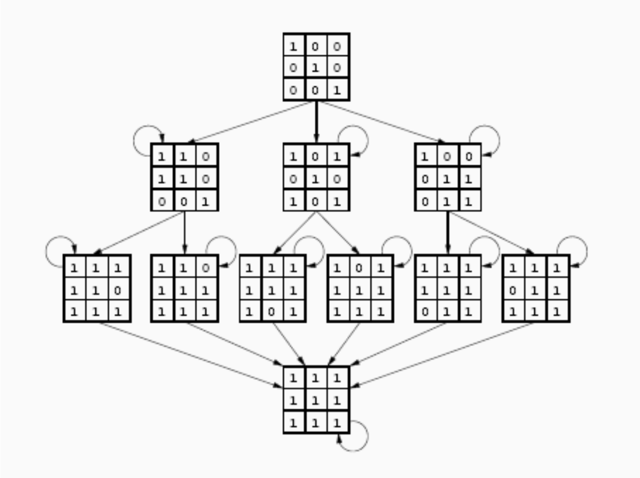

Figure 1: Matrix model of gossiping girls [10]

(i) Smaller local view sizes; in the gossiping girls example it is implicitly assumed that each girl can memorize each gossip (i.e., c=n), but as in a distributed system the amount of local memory is typically limited, this is no longer adequate.

(ii) A probabilistic mechanism to select the node to send the gossip(s) to. Each node randomly selects the set of nodes to which it sends its own information.

This yields the matrix $\widehat{\mathbf{G}}$ which stores for each node i the *probability* $\widehat{\mathbf{G}}(i, j)$ that this node has view j. Note that a view is a set of information items, e.g., a set of node identities or a set of measured data, etc.. The resulting state space has "only" $\binom{n-1}{c} \cdot n$ states, as opposed to the explicit modeling approach which yields $\binom{n-1}{c}^n$ states.

Clearly, in this approach the state transitions —how does one matrix evolve into another one?— are more involved than in the gossiping girls example. Krieger *et al.* [8] describe in detail how the matrices evolve. They conjecture that this evolution stabilizes under some mild conditions and that the stable situation is independent from the initial matrix. This strongly resembles the existence of a steady state in a Markov chain. A formal proof though is not given and is an interesting subject for further investigation. Interestingly, the authors also show how spontaneous joins and leaves of nodes can be accommodated by modeling nodes as being either active or passive, and model the durations of being active (or passive) as geometric distributions. In addition, aging is incorporated allowing for a more natural modeling of old and recent data.

Counting abstraction. At first sight, Bakhshi *et al.* [2] take a similar approach as [4, 9] as they model each node's behaviour by a Markov chain. The global state of the system is, however, not composed of each individual state of each node, but rather an abstraction is taken where one only keeps track of how many nodes (in fact, the fraction of nodes) that are in a certain state. This approach is appealing as the state space of each node is rather small and is referred to as counting abstraction in the field of formal verification, see e.g., [1, Chapter 7]. This yields a state space of just $n+1$ states, i.e., a state space that is linear in the number of nodes. The nodes are assumed to proceed in a lock-step fashion, i.e., each node is assumed to perform a (possible idle) step in each global step. This synchronous parallel composition is conceptually different from the asynchronous and random composition of [9] and [4] respectively. The local transition probabilities of a node depend on the current state of the node, as well as on the fraction of nodes in certain states—the so-called occupancy measure. Under mild conditions, the occupancy measure almost surely converges when the number of nodes goes to infinity. Moreover, this limit can be computed by an easy recursive scheme. Bakhshi *et al.* [2] explain the details of this modeling approach by means of a simple example, and apply it to the gossiping time protocol of Iwanicki *et al.* [6] for a static network topology. This modeling approach is novel, and rather appealing due to its simplicity both in modeling as well as in analysis (see below). It would be interesting though to see how this approach can be adapted to dynamic network topologies.

3. ANALYSIS

It remains to discuss the analysis techniques. Crouzen *et al.* [4] and Kwiatkowska *et al.* [9] both use a rather new and promising analysis technique: probabilistic model checking. A detailed account and introduction to this technique can be found in, e.g., [1, Chapter 10] and in an earlier issue of this journal [5]. This technique has been successfully applied to case studies from a broad rannge of applications, such as security, randomized distributed algorithms, and systems biology. The main advantage of this approach is that it is fully automated, i.e., measures such as "what is the probability that all girls know all gossips after 5 phone calls?" can be determined using software tools such as PRISM and CADP. [2]

The same tools can be used for a rich class of other measures. Detailed knowledge about the analysis techniques is not needed, which is a big advantage for protocol designers. The main result of Crouzen *et al.* [4] is that their analysis results confirm the simulation results published in the literature. On the other hand, the analysis of Kwiatkowska *et al.* [9] shows that there is a considerable discrepancy between minimal and maximal probabilities, and moreover, that average values are sometimes far off these extreme values. This is in my opinion a strong argument for favouring asynchronous modeling of concurrency. Network sizes in both papers are somewhat disappointing; the largest system instance has 7 nodes and a local view of size 2. This is still far away from realistic sizes of decentralized systems and incomparable to network sizes that can be dealt with using simulation. On the positive side, phenomena that can be assessed for small-size system instances can be extremely helpful to improve the understanding of the protocol's functioning, to judge parameter settings, and thus to steer design decision. To bridge the gap between the currently tractable state space sizes and those occurring in practice, is a major challenge for the coming years.

[2]The solution to instances of the gossiping girls example, e.g., can easily be computed using off-the-shelf model checkers in a few seconds.

The appealing matrix modeling approach yields rather compact models, yet analysis remains a major open issue. The authors report on a prototypical implementation that mimics the evolution of the matrices. Experiments indicate that indeed the matrices converge. For a configuration with n=6 and c=2, the computation for a static network without aging takes about 30 iterations. It remains an open problem though how to obtain measures-of-interest from the matrix representations. This is a challenging issue for further research.

Bakhshi *et al.* [2] resort to results from the area of mean-field analysis, a technique that originates from physics. The main foundation result is that for n to infinity, the stochastic process representing the system's behaviour converges to a limit that can be easily computed and is far less expensive to compute than using, e.g., simulation or emulation [3]. The authors compare simulative results with results from mean-field analysis for a gossiping protocol that aims to establish clock synchronization. For networks of up to 2,000 nodes relatively accurate results are obtained. Due to the nature of the mean-field theorem, the larger the number of nodes, the more accurate the results. The relative difference is at most 20%, and in most (larger system instances) cases less than 1%.

4. CONCLUSION

This ACM PER issue does not present a final report on full-fledged techniques to handle gossiping protocols. Not at all. Instead, the reader should consider this issue as a rich source of encouraging papers with many challenges for further research ranging from scaling up state-based analysis techniques to the development of new modeling and analysis techniques tailored to gossiping protocols, such as the appealing matrix modeling approach and the mean-field analysis technique. Clearly, the dominating modeling and analysis technique has not been found yet, and extensive research is required to obtain powerful and adequate techniques. The main issue seems to be able to deal with the excessive number of nodes that participate in gossiping protocols. Although it is unclear where to look for the silver bullet (and beat simulation), one thing is evident from this issue: modeling and analysis of gossiping protocols is an enormously challenging field of research!

5. REFERENCES

[1] C. Baier and J.-P. Katoen. *Principles of Model Checking.* MIT Press, 2008.

[2] R. Bakhshi, L. Cloth, W.J. Fokkink, and B. R. Haverkort. Mean-field analysis for the evaluation of gossip protocols. This issue, 2008.

[3] J.-Y. le Boudec, D. McDonald, and J. Mundinger. A geberic mean field convergence result for a system of interacting objects. Quantitative Evaluation of Systems (QEST), pp. 3–18, IEEE CS Press, 2007.

[4] P. Crouzen, J. van de Pol, and A. Rensink. Applying formal methods to gossiping networks with mCRL and Groove. This issue, 2008.

[5] B. R. Haverkort and J.-P. Katoen (eds). Performance and verification. *ACM SIGMETRICS Performance Evaluation Review* **32**(4), 2005.

[6] K. Iwanicki, M. van Steen, and S. Voulgaris. Gossip-based clock synchronization for large decentralized networks. *IEEE Int. Workshop on Self-Managed Networks, Systems and Services*, pp. 28–42, 2006.

[7] M. Jelasity, S. Voulgaris, R. Guerraoui, A.-M. Kermarrec, and M. van Steen. Gossip-based peer sampling. *ACM TOCS* **25**(3):8, 2007.

[8] T. Krieger, M. Riedl, J. Schuster and M. Siegle. A view-probability-matrix approach to the modelling of gossip protocols. This issue, 2008.

[9] M. Z. Kwiatkowska, G. Norman, and D. Parker. Analysis of a gossip protocol in PRISM. This issue, 2008.

[10] F. W. Vaandrager. Wat is model checking? Radboud University Nijmegen, 2008 (in Dutch).