Formal Analysis Techniques for Gossiping Protocols

Rena Bakhshi
Vrije Universiteit Amsterdam
Amsterdam, Netherlands
rbakhshi@few.vu.nl

Francois Bonnet
ENS Cachan/IRISA
Rennes, France
fbonnet@irisa.fr

Wan Fokkink
Vrije Universiteit Amsterdam
Amsterdam, Netherlands
wanf@cs.vu.nl

Boudewijn Haverkort
University of Twente
Enschede, Netherlands
brh@cs.utwente.nl

ABSTRACT

We give a survey of formal verification techniques that can be used to corroborate existing experimental results for gossiping protocols in a rigorous manner. We present properties of interest for gossiping protocols and discuss how various formal evaluation techniques can be employed to predict them.

Categories and Subject Descriptors

A.1 [**Introductory and survey**]; C.2 [**Computer-communication Networks**]; C.2.2 [**Network Protocols**]: Protocol verification; C.2.4 [**Distributed Systems**]: Distributed applications; I.6.4 [**Model Validation and Analysis**]

General Terms

Design, Theory, Verification.

Keywords

Gossiping protocols, verification techniques, survey.

1. INTRODUCTION

The emergence of the Internet as a computing platform asks for new classes of algorithms that combine massive distributed processing and inherent decentralization. These algorithms should be able to execute in an environment that is heterogeneous, changes almost continuously, and consists of millions of nodes. Massive parallel computing on the Internet also demands a degree of self-organization; the amount of devices and software is simply too large to be managed by humans.

Gossiping protocols have shown to be a sensible paradigm for developing stable and reliable communication mechanisms that scale up to massively parallel environments. In a gossiping (also called epidemic) protocol, nodes exchange data similar to the way a contagious disease spreads. That is, a participating peer can select, according to some probability distribution, other peers to exchange information with. Gossiping protocols were originally applied in database replication [26], but more recently also for failure detection [70], and resource monitoring [69]. They are employed in wired as well as wireless environments. In a massively parallel setting, the gossip mechanism should be used at very high speeds, yielding a new generation of protocols that have an unusual style of probabilistic reliability guarantees, regarding scalability, performance, and stability of throughput. Surveys [30, 32, 47] provide an introduction to the field.

Gossiping protocols tend to contain several design parameters that can influence the non-functional properties of these protocols, e.g., performance, robustness and fault tolerance. Values of these parameters are usually determined empirically, without a proper understanding why the protocol performs well for these values, and without any certainty that these values are close to optimal or robust choices. Thorough experimental analysis in [49] has shown that the emergent behaviour of gossiping protocols may vary substantially by changing only a few design parameters.

When a large number of programs interact in a connected environment, various phenomena occur that are not explicable in terms of the behaviour of any single agent. It is necessary to understand these phenomena in order to keep the overall systems both stable and efficient. Distributed algorithms and protocols that run steadily and reliably in small-scale settings, tend to lose those properties as numbers of users, the size of the network and transaction processing rates all increase. Typical problems are disruptive overloads and routing changes, periods of poor connectivity and throughput instability. Failures rise in frequency simply because the numbers of participating components are larger [71].

In practice, properties of gossiping protocols are usually diagnosed by emulating such systems. However, in principle, owing to their often relatively simple structure, gossiping protocols lend themselves very well to formal analysis, in order to predict their behaviour with high confidence. A complication in the analysis of gossiping protocols is that they are meant to work in very large networks, and for ad hoc wireless networks even with lossy channels. In this paper we give an overview of the different approaches that can be taken to formally analyse gossiping protocols, and which properties of such protocols can be verified with which formal verification techniques.

This paper is structured as follows. Section 2 gives an overview of the different types of requirements for gossiping protocols. Section 3 presents the available spectrum of analysis techniques. Finally, Section 4 contains some conclusions.

2. REQUIREMENTS

Requirements for gossiping protocols can be divided into three classes: general, functional and non-functional requirements. These will be discussed in the current section. We use terminology from [49, 72].

2.1 General Requirements

Gossiping protocols satisfy the following general requirements:

- *Simplicity:* The protocol is simple and easy to deploy. For example, in a wireless network, a node should be able to join the system without executing a complex procedure ("plug-and-play").
- *Scalability:* Each node continues to perform its operations at almost the same rate irrespective of the network size. For example, the local knowledge (neighbours list) of a node does not increase with the network size.
- *Symmetry:* In a large-scale network, all nodes play identical roles. Hence, there is no single point of failure. Randomization, e.g., random peer selection, tends to fit into this requirement, because each node typically runs the same algorithm.

2.2 Functional Requirements

Functional requirements describe properties of the outputs of a system, and how a certain input is transformed into an output. We classify several functional requirements on gossiping protocols. We distinguish between global and local properties.

1. Global properties of the system:
   - *Connectivity:* This can, for example, be expressed as a minimum number of links between nodes, whose removal will result in the partitioning of the network graph. The connectivity of a graph is an important measure of its robustness, because partitioning of a network graph creates difficulties for information dissemination.
   - *Convergence:* One can distinguish between convergence of the system parameters to some values (e.g., achieving a certain accuracy in the estimates of the aggregate function values) and convergence of the system structure to some particular type of graph.
2. Local properties of nodes:
   - *Degree distribution:* The degree of a node is the number of its neighbours in the network graph. This concept is interesting because of its relationship to robustness of a graph in the presence of node failures, its effect on patterns of epidemic spread, and its importance in the distribution of resource usage of nodes.
   - *Clustering coefficient:* The clustering coefficient of a node expresses a ratio of the number of links between the node's neighbours to the number of all possible links between them. Intuitively, it shows how many neighbours of a node are neighbours among themselves. Analysis of this property is interesting because a high clustering coefficient affects information dissemination, as the number of redundant messages increases. Also, it affects the self-healing capacity, by strengthening the connectivity within a cluster, thus decreasing the probability of partitioning.
   - *Shortest path length:* The shortest path length between two nodes is the minimum number of edges that must be traversed to go from one node to the other. The average path length is the average of all shortest path lengths between any two nodes in a graph. The shortest and average path length give information on the time and communication costs to reach a node.

2.3 Non-Functional Requirements

Non-functional requirements regard the quality (e.g., performance, maintainability, fault-tolerance) and economics (e.g., timing, cost) of system behaviour. The following high-level non-functional requirements can be identified for gossiping protocols:

- *Time Complexity:* The number of time units it takes (at worst or on average) for a gossiping protocol to "infect" every node in the network, e.g., for data delivery to all nodes, or for computation of an aggregation function output.
- *Message Complexity:* The total number of gossiping messages (at worst or on average) exchanged over the network during an execution.
- *Information dissemination:* There should be a high probability that a piece of information is shared with all processes within a given time.
- *Robustness:* The ability of a gossiping protocol to maintain correct system operation in the face of massive node crashes and node churn.
- *Graceful degradation:* Large numbers of node failures in the system may affect its operation. However, performance, functionality and reliability of gossiping protocols should not drop rapidly as the number of failures increases.
- *Elasticity:* Robustness of the well-operation of the systems in face of largely varying node capabilities in terms of memory, bandwidth and connectivity.

- *Self-organization:* The nodes should be able to organize themselves in unpredictable circumstances without external interventions. For example, in gossiping protocols a network graph forms overlays that are adaptable to network and environmental changes.

Specifically in wireless networks, nodes communicate through error-prone radio channels and typically also have limited computational capabilities. Special design issues then include energy use, mobility, transmission power, memory usage and latency. Network properties such as message reliability and node reachability may in that case influence the behaviour of the protocol.

3. FORMAL ANALYSIS TECHNIQUES

The aims when analysing a system are in general to obtain a better understanding of and gain further confidence in the system's behaviour, to detect possible errors, and to improve its design. A complication in the analysis of gossiping protocols is that they are meant to work in very large networks. Properties of such protocols are generally diagnosed by emulating such networks.

The formal specification of systems helps to make explicit the underlying assumptions (like the synchronization primitives), which tend to remain hidden in implementations or simulation exercises. Also such a specification can be analysed using (semi-)automated formal verification techniques. There is a rich history of the use of such techniques for verifying a variety of desirable properties for a wide range of systems. The efficiency of a particular formal analysis technique depends on the system under study.

Gossiping protocols in general contain several design parameters that heavily influence their behaviour. For example, the number of protocol cycles, message forwarding strategies, message delays, or cache usage and size. Formal analysis techniques can help in the search for optimal values of such parameters.

Rigorous formal analysis techniques for gossiping protocols have so far hardly been applied in large-scale settings, and need to be developed further for this purpose. The main aim of this paper is to investigate which formal analysis techniques can in principle be employed efficiently in the analysis of gossiping protocols, for wired as well as wireless systems. We will provide an overview of the existing analysis methods, their use and limitations, as well as a comparison of related work on the formal verification of gossiping protocols. Our aim is to create a better understanding for selecting a suitable approach for such an analysis.

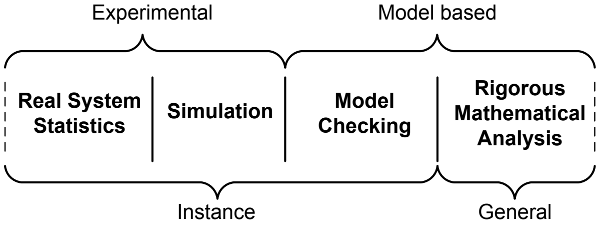

Figure 1: Spectrum of validation

Figure 1 depicts the spectrum of analysis techniques, ranging from experimental work with a real system implementation up to rigorous mathematical analysis. Real system statistics and simulation techniques are based on experiments performed on the system and on collecting data statistics either from the running system through monitoring it at real time or from a discrete-event simulation of the system. Usually, these approaches are used to study the behaviour of a particular implementation (instance) of the system. The other approaches require a formal modelling of the system. Typically, they are used to verify specific properties in a more general context. Although these methods often require particular assumptions to be made, their advantage is that they can be used before a system is being implemented, and that in principle they are not costly (in comparison to full-scale experiments on a real system).

In the following subsections, we present pros and cons of the use of experimental and model-based formal analysis techniques for gossiping protocols.

3.1 Experimental Evaluation

In practice, properties of gossiping protocols are often diagnosed by emulation, and through performing simulations (see, e.g., [36]). Commonly used discrete-event simulators are ns-2, Opnet and Glomosim; but often a customized simulator is built in for example Java or Matlab.

Experimentation is the major source of gaining first insight. The reason is that in reality performance of gossiping protocols depends on many factors: characteristics of the network, certain distributions (capacity, node-degree, etc), usage scenarios, user models, incentives, etc.

However, different simulators can produce vastly different results, even for simple systems (see, e.g., [18]). The reason is that simulators employ different models for the medium access control and physical layers. The results of the simulators say as much about the simulated system as about the particular lower-level implementation of the simulator. Also the employed random number generators have an (unpredictable) impact; for instance, [2] questions the credibility of this type of simulations. Moreover, different simulation analyses of gossiping protocols make different assumptions about the underlying model, which makes comparison of results difficult. For example, [18] and [19] both evaluate the flooding protocol, but sending and receiving is perfectly synchronized in [18], while [19] assumes a random waiting period between sending and receiving.

Surprisingly, few attempts have been made to implement systems that use one of the existing gossiping protocols; we are only aware of Astrolabe [69], Tribler [64] and ARRG [28]. Notably, in [28], it is shown that a gossiping protocol that behaves well in an emulated environment, may not behave well when truly implemented, especially in an environment where nodes can crash.

3.2 Model-Based Analysis Techniques

The formal specification of a system helps to obtain not only a better (more modular) description, but also a clear understanding and an abstract view of the system. Formal analysis techniques, typically referred to as formal verification, are supported by (semi-)automated tools. They can detect errors in the design that are not so easily found using emulation or testing, and can be used to establish the correctness of the design. The most effective way to apply formal methods is actually during the design of a system, rather than after-the-fact, as is, unfortunately, often the case.

Formal models need to be realistic. An experimental evaluation can help to obtain a first insight in the behaviour of a system, and to identify which characteristics need to be included into the model.

There are two main approaches to formal verification. The first approach involves a rigorous mathematical analysis of the properties of the system, using results from for instance calculus and probability theory. Such an analysis can be supported semi-automatically by means of Matlab or a theorem prover. While Matlab is an easy to use but imprecise mathematical tool, a theorem prover requires a lot of effort from the user but supports precise mathematical reasoning about the system. Important theorem proving tools are Isabelle/HOL [61], PVS [63] and Coq [11].

The second approach is model checking, which consists of a systematic and fully automatic exploration of the state-space of the system specification. The explorative nature of the approach in principle requires that the state space is finite. However, recent work also addresses symbolic model checking techniques for infinite-state models.

3.2.1 Rigorous Mathematical Analysis

Rigorous analysis techniques for gossiping protocols are built on sound mathematical foundations, and draw inspiration from the mathematical theory of epidemics [30, 6]. These analysis techniques are in general used to verify specific properties of a gossiping protocol. Therefore, such a study is usually done on a simplified system model of the actual protocol: one has to decide which characteristics of the protocol should be studied (see Section 2), and which parameters of the protocol should be modelled in order to study these characteristics.

Gossiping protocols are intrinsically probabilistic. For instance, a node may randomly selects a "gossip" partner or a data item for the exchange with another node. Or it may be the case that when a node receives a message, then with some probability p it forwards the message to all (or some) of its neighbours, while with probability $1 - p$ the message is purged. A key property is that if the probability p is sufficiently large, and the network sufficiently dense, then the probability of successful information spread remains close to 1, while the number of sent messages is relatively small compared to flooding.

Thus, the mathematical foundations underlying the modelling and verification of gossiping protocols can be found in probability theory.

Hand-crafted Markov chains

Markov chains can be used for modelling a variety of aspects of gossiping protocols. Markov chains allow to capture the evolution of gossip-based systems; each state of the Markov chain describes one state of the system. However, a state of the Markov chain does not represent a state of the whole system, but only a state of the system limited to the list of parameters that are modelled. From one state of the Markov chain to another, there are probabilistic transitions corresponding to the probabilistic evolution of the system. There are two types of Markov chain, distinguished by the transitions occurring at any time (continuous-time Markov chain) or only for defined steps (discrete-time Markov chain). By analysing the different possible sequences (and their probabilities), it is possible to obtain a global insight in the operation of the protocol. The Markov chain describing the system evolution converges to a useful distribution over all possible system states, from which interesting protocol properties can be concluded. For more information, we refer to [42, 66].

To exemplify the approach in the context of gossiping protocols, we provide the description of two case studies: the first one concerns the degree distribution of nodes, and the second the connectivity of network overlays. In both examples, the results from the mathematical analysis have been compared to the results of simulations to confirm their validity.

Bonnet [13] studied the evolution of the in-degree distribution of nodes during the execution of the Cyclon protocol [73]. Markov chains model this distribution, that is, the probability of being in state i of the Markov chain equals the probability for a given node to have i in-edges. From the designed Markov chain it is possible to determine the stationary distribution of the in-degrees, i.e., the distribution to which the protocol converges, as well as to calculate bounds on convergence time.

Allavena *et al.* [1] proposed a scalable gossip-based algorithm for local view maintenance. They counted the number of links between two parts of the system (say A and B) and studied the evolution of these numbers; the states of the associated Markov chain are the numbers of links from A to B and from B to A. From the designed Markov chain the expected time until a network partition occurs was calculated. This case study also included a model of the system under churn.

As other examples of the use of Markov chains for gossiping and related protocols, we refer to studies on gossip-based membership management [1, 7, 13], gossip-based distributed aggregation [15, 16, 60, 27, 53], gossip-based information dissemination [23, 29] and network topology change [34]. Schnoebelen [67] surveyed several proposals for modelling probabilistic lossy channel systems with Markov chains and the verification techniques they support. It would be interesting to see whether these ideas can be of use in both quantitative and qualitative analyses of gossiping protocols in wireless networks. Probability theory has further been applied to analyse gossip-based information dissemination [35, 50, 12, 58] and gossip-based resource location [54, 55].

Mathematical analysis can be combined with simulations to validate the results and understand the system behaviour. A strong point of mathematical analysis is that often it scales well with respect to the size of a network. However, it requires a lot of effort, it can only be used to analyse a limited class of properties, and the assumptions that are invariably made to simplify the analysis often affect the accuracy of the results [14]. For example, the analyses in [50, 55, 54] rely on the assumption of full knowledge of group membership, ignoring its practical infeasibility.

3.2.2 Model Checking

Model checking is an exhaustive state space exploration technique that is used to validate formally specified system requirements with respect to a formal system description [21]. Such a system is verified for a fixed configuration; so in most cases, no general system correctness can be obtained.

Using some high-level formal modelling language, automatically an underlying state space can be derived, be it implicitly or symbolically. The system requirements are specified using some logical language, like LTL, CTL or extensions thereof [48]. Well-known and widely applied model checking tools are SPIN [46], Uppaal [8] (for timed systems), and PRISM [45] (for probabilistic systems). The system specification language can, e.g., be based on process algebra, automata or Petri nets.

Model checking suffers from the so-called state explosion problem, meaning that the state space of a specified system grows exponentially with respect to its number of components. The main challenge for model checking lies in modelling large-scale dynamic systems. To overcome the state explosion problem and to speed up the verification process, various state space reduction techniques have been proposed. For instance, combinations of symbolic verification techniques with explicit state space exploration (symbolic model checking), verification of properties on a smaller abstract model of the system under scrutiny, possibly obtained after bisimulation reduction, parallelisation of verification algorithms, partial exploration of the state space (bounded model checking, on-the-fly model checking, partial order reduction), and efficient state representation (bitstate hashing), have been proposed to make model checking practically feasible.

Initial model checking approaches used as underlying mathematical model a finite-state automaton, i.e., a model with neither explicit time nor probabilities. Recently, model checking techniques have been proposed for system models including both time and probabilities, possibly in combination with non-determinism. In view of the probabilistic features in gossiping protocols, we focus on model checking of probabilistic models.

Probabilistic and Stochastic Model Checking

In probabilistic and stochastic model checking, the underlying system model is not represented by an automaton, but instead as a stochastic process of some sort, mostly a finite-state Markov chain (discrete- or continuous-time). Often, these Markov chains are extended with state labels and transition labels (so-called action names). These Markov chains are mostly specified using some high-level formalism, like stochastic process algebra [20] or stochastic Petri nets.

Gossiping protocols may require models which, in addition to pure probabilistic choices, also allow for non-deterministic choices. That is, it is possible in a given state of a system to non-deterministically move to another state with some probability. Here, Markov decision processes can be applied [65]. The key idea to a Markov decision process is to allow a set of probability distributions in each state instead of a single distribution as in Markov chains. The choice between these distributions is made externally and non-deterministically, either by a scheduler that decides which sequential subprocess takes the next step (as in e.g., concurrent Markov chains), or by an adversary that influences or affects the system. Probabilistic choices are internal to the process and made according to the selected distribution.

The system requirements of interest are again specified through logical expressions, over the paths that can be taken through the model. For that purpose, the logics are extended to include a notion of time and probability. Prominent examples of such logics are CSL [5, 4], CSRL [41] and asCSL [3] for continuous-time models, and pCTL [40] for discrete-time models.

Where traditional model checking algorithms lean heavily on determining reachability of certain states or state groups (or the non-reachability), in probabilistic and stochastic model checking also the time until some states are reached (or avoided) plays a major role. Furthermore, reachability of states is expressed in terms of a probability (no mutually exclusive yes or no) and a time-bound; as an example, certain states might be highly probably reached for short time periods, but not for longer time periods. Stochastic model checking relies on algorithms for reachability analysis, as well as on numerical algorithms for determining long-term and transient behaviour in Markov chains. Although such algorithms are well understood, their implementation requires care, especially if very large models are to be addressed.

Stochastic and probabilistic model checking has been applied in a wide variety of case studies, ranging from workstation cluster availability [43] to the evaluation of power-saving methods [62] and the analysis of wireless (sensor) networks [59].

We feel that the success of model checking approaches, and especially stochastic and probabilistic model checking approaches, to a wide variety of case studies, is promising. This observation is also fuelled by the fact that gossiping protocols with their probabilistic and asynchronous behaviour fit well to the model classes supported by the known model checking techniques. This is not to say that we do not expect difficulties. On the contrary, the key to successfully verifying gossip-based systems lies in coping with their scale. This implies that an analysis or verification technique should be able to deal with large networks somehow, be it through smart abstractions or approximations (thus avoiding large state spaces), or through smart storage techniques or brute force distributed verification algorithms.

Some of the optimization techniques for general modelling techniques, as described earlier, have been adapted to probabilistic model checking, in particular: abstraction [52, 56, 57, 17], distribution [9, 10, 38] and Markovian bisimulation [51].

Approximate and Statistical Probabilistic Model Checking

An alternative approach to cope with state explosion for probabilistic systems is found in approximate probabilistic model checking. The main idea of this approach is to apply Monte-Carlo sampling techniques [39, 33]; the resulting probabilities are accurate only with respect to some accuracy criterion.

Approximate probabilistic model checking [44, 38] is an approximation method for the logic restricted to time-bounded safety properties ("positive" LTL). Monte Carlo model checking [37] is based on a randomized algorithm for probabilistic model checking of safety properties for general LTL model checking; Monte Carlo model checking uses the optimal approximation algorithm of [22].

In so-called statistical probabilistic methods (e.g., [74]), statistical hypothesis testing is used instead of randomized approximation schemes. The approach of [75] describes a model-independent procedure for verifying properties of discrete-event systems based on Monte-Carlo simulation and statistical hypothesis testing. The procedure uses a refinement technique to build statistical tests for the satisfaction probability of CSL formulas. The statistical method of [68] concentrates on model checking of black-box probabilistic systems against specifications given in a sub-logic of CSL.

Similar to the idea of approximation-based probabilistic model checking, [31] combines probabilistic model checking with Monte Carlo simulations for the performance analysis of probabilistic broadcast protocols in a wireless network. In particular, this study shows results for reliability and reachability properties under different assumptions, such as message collision, lossy channels and unreliable timing, and their impact on the results.

Case study [24] presents the modelling of a sensor network using approximate probabilistic model checking. Another case study [17] presents the results of an analysis of the MAC protocol for sensor networks using approximate probabilistic model checking. eXtended Reactive Modules (XRM) [25] have been proposed for modelling wireless sensor networks to generate RM models suitable for PRISM and approximate probabilistic model checking.

4. CONCLUSIONS

Concluding, the formal analysis of gossiping protocols is a rather unexplored research field, with many challenges and open problems ahead. A more insightful and systematic methodology should be developed, that targets gossiping protocols. The assumptions made for simplifying such an analysis should be restricted as much as possible, as otherwise the analysis itself becomes unrealistic.

Markov chains give a precise mathematical description, but the analysis is time-consuming and can only be used for a restricted class of properties. It would be worthwhile to use theorem proving tools in order to support such a mathematical analysis.

Probabilistic model checking techniques are convenient to use, as they are based on verification algorithms. But formally modelling a gossiping protocol still requires considerable effort, and can introduce mistakes by itself (if the model deviates from the actual protocol). Also the verification algorithms are very much under development, and probabilistic model checking is, even more than standard model checking, suffering from the state explosion problem. This complicates the analysis of gossiping protocols considerably, as they are supposed to be applied in large-scale networks. The use of optimisation techniques, like abstraction and distributed verification, will form important ingredients for model checking to become practically of interest for the evaluation of gossiping protocols.

Approximate probabilistic model checking could serve as a good compromise between probabilistic model checking and simulation. They do not provide an exhaustive search to verify a given property, and as a result they do not suffer from the state explosion problem that much. Still in practice they can provide rather accurate probabilistic estimates. But approximate probabilistic model checking is still coming off age, and needs to be further developed.

5. ACKNOWLEDGMENTS

The authors would like to thank Joost-Pieter Katoen and the anonymous reviewers for helpful comments.

6. REFERENCES

[1] A. Allavena, A. Demers, and J. E. Hopcroft. Correctness of a gossip based membership protocol. In *Proc. 24th Annu. ACM Symp. on Principles of Distributed Computing (PODC'05)*, pages 292–301. ACM Press, 2005.

[2] T. Andel and A. Yasinsac. On the credibility of MANET simulations. *IEEE Computer*, 37(7):48–54, 2006.

[3] C. Baier, L. Cloth, B. R. Haverkort, M. Kuntz, and M. Siegle. Model checking markov chains with actions and state labels. *IEEE Trans. Software Eng.*, 33(4):209–224, 2007.

[4] C. Baier, B. R. Haverkort, H. Hermanns, and J.-P. Katoen. Model Checking Continuous-Time Markov Chains by Transient Analysis. In *Proc. 12th Int. Conf. on Computer Aided Verification (CAV'00)*, volume 1855 of *LNCS*, pages 358–372. Springer, 2000.

[5] C. Baier, B. R. Haverkort, H. Hermanns, and J.-P. Katoen. Model-checking algorithms for continuous-time Markov chains. *IEEE Trans. Software Eng.*, 29(6):524–541, 2003.

[6] N. T. Bailey. *Mathematical Theory of Infectious Diseases and Its Applications.* Griffin, London, second edition, 1975.

[7] Z. Bar-Yossef, R. Friedman, and G. Kliot. RaWMS: random walk based lightweight membership service for wireless ad hoc network. In *Proc. 7th ACM Int. Symp. on Mobile ad hoc networking and computing (MobiHoc'06)*, pages 238–249. ACM Press, 2006.

[8] G. Behrmann, A. David, and K. G. Larsen. A tutorial on UPPAAL. In *Formal Methods for the Design of Real-Time Systems: Proc. 4th Int. School on Formal Methods for the Design of Comput., Commun. and Software Syst. (SFM-RT 2004)*, number 3185 in

LNCS, pages 200–236. Springer, 2004.

[9] A. Bell and B. R. Haverkort. Sequential and distributed model checking of Petri nets. *Software Tools for Technology Transfer (STTT)*, 7(1):43–60, 2005.

[10] A. Bell and B. R. Haverkort. Distributed disk-based algorithms for model checking very large Markov chains. *Formal Methods in System Design*, 29(2):177–196, 2006.

[11] Y. Bertot and P. Castéran. *Interactive Theorem Proving and Program Development Coq'Art: The Calculus of Inductive Constructions*, volume XXV of *Texts in Theoretical Computer Science: An EATCS Series*. Springer, 2004.

[12] K. Birman, M. Hayden, O. Ozkasap, Z. Xiao, M. Budiu, and Y. Minsky. Bimodal multicast. *ACM Trans. on Comput. Syst.*, 17(2):41–88, 1999.

[13] F. Bonnet. Performance analysis of Cyclon, an inexpensive membership management for unstructured p2p overlays. Master thesis, ENS Cachan Bretagne, University of Rennes, IRISA, 2006.

[14] J. Bowen and M. Hinchey. Ten Commandments of Formal Methods ...Ten Years Later. *Computer*, 39(1):40–48, 2006.

[15] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Gossip algorithms: Design, analysis and applications. In *Proc. 24th Conf. of the IEEE Comput. and Commun. Soc. (INFOCOM 2005)*, volume 3, pages 1653–1664. IEEE Press, 2005.

[16] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Mixing times for random walks on geometric random graphs. In *Proc. 7th Workshop on Algorithm Eng. and Experiments and 2nd Workshop on Analytic Algorithmics and Combinatorics (ALENEX/ANALCO 2005)*, pages 240–249. SIAM, 2005.

[17] M. Cadilhac, T. Hérault, R. Lassaigne, S. Peyronnet, and S. Tixeuil. Evaluating complex MAC protocols for sensor networks with APMC. In *Proc. 6th Int. Workshop on Automated Verification of Critical Syst (AVoCS'06)*, ENTCS. Elsevier, 2006. To appear.

[18] R. Cardell-Oliver. Why flooding is unreliable in multi-hop, wireless networks. Technical Report UWA-CSSE-04-001, University of Western Australia, 2004.

[19] D. Cavin, Y. Sasson, and A. Schiper. On the accuracy of MANET simulators. In *Proc. 2nd ACM Int. Workshop on Principles of Mobile Computing (POMC'02)*, pages 38–43. ACM Press, 2002.

[20] A. Clark, S. Gilmore, J. Hillston, and M. Tribastone. Stochastic process algebra. In *Formal Methods for the Performance Evaluation: Proc. 7th Int. School on Formal Methods for the Design of Comput., Commun. and Software Syst. (SFM-PE 2007)*, number 4486 in LNCS, pages 132–179. Springer, 2007.

[21] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking.* MIT Press, 2000.

[22] P. Dagum, R. Karp, M. Luby, and S. Ross. An optimal algorithm for monte carlo estimation. *SIAM J. Comput.*, 29(5):1484–1496, 2000.

[23] S. Deb, M. Médard, and C. Choute. Algebraic gossip: a network coding approach to optimal multiple rumor mongering. *IEEE/ACM Trans. Netw.*, 14(SI):2486–2507, 2006.

[24] A. Demaille, S. Peyronnet, and T. Hérault. Probabilistic verification of sensor networks. In *Proc. 4th IEEE Int. Conf. on Comput. Sci., Research, Innovation and Vision for the Future (RIVF'06)*, pages 45–54. IEEE Computer Society, 2006.

[25] A. Demaille, S. Peyronnet, and B. Sigoure. Modeling of sensor networks using XRM. In *Proc. 2nd Int. Symp. on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA'06)*, 2006.

[26] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry. Epidemic algorithms for replicated database maintenance. In *Proc. 6th Annu. ACM Symp. on Principles of Distributed Computing (PODC'87)*, pages 1–12. ACM Press, 1987.

[27] A. G. Dimakis, A. D. Sarwate, and M. J. Wainwright. Geographic gossip: efficient aggregation for sensor networks. In *Proc. 5th Int. Conf. on Inform. processing in sensor networks (IPSN'06)*, pages 69–76. ACM Press, 2006.

[28] N. Drost, E. Ogston, R. van Nieuwpoort, and H. Bal. ARRG: Real-World Gossiping. In *Proc. 15th IEEE Int. Symp. on High Performance Distributed Computing (HPDC-15'06)*. IEEE Computer Society, 2007. To appear.

[29] P. Eugster, R. Guerraoui, S. B. Handurukande, P. Kouznetsov, and A.-M. Kermarrec. Lightweight probabilistic broadcast. *ACM Trans. on Comput. Syst.*, 21(4):341–374, 2003.

[30] P. Eugster, R. Guerraoui, A.-M. Kermarrec, and L. Massoulié. From epidemics to distributed computing. *IEEE Computer*, 37(5):60–67, 2004.

[31] A. Fehnker and P. Gao. Formal verification and simulation for performance analysis for probabilistic broadcast protocols. In *Proc. 5th Conf. on Ad-Hoc, Mobile, and Wireless Networks (ADHOC-NOW'06)*, volume 4104 of *LNCS*, pages 121–141. Springer, 2006.

[32] U. Feige, D. Peleg, P. Raghavan, and E. Upfal. Randomized broadcast in networks. In *SIGAL Int. Symp. on Algorithms*, volume 450 of *LNCS*, pages 128–137. Springer, 1990.

[33] G. S. Fishman. *Monte Carlo: Concepts, Algorithms, and Applications.* Springer Series in Operations Research. Springer, 1996.

[34] A. Ganesh, L. Massoulie, and D. Towsley. The effect of network topology on the spread of epidemics. In *Proc. 24th of the IEEE Comput. and Commun. Soc. (INFOCOM 2005)*, volume 2, pages 1455–1466. IEEE Computer Society, 2005.

[35] A. J. Ganesh, A.-M. Kermarrec, and L. Massoulie. SCAMP: Peer-to-peer lightweight membership service for large-scale group communication. In *Proc. 3rd Int. Workshop on Networked Group Commun. (NGC'01)*, volume 2233 of *LNCS*, pages 44–55. Springer, 2001.

[36] D. Gavidia, S. Voulgaris, and M. van Steen. A Gossip-based Distributed News Service for Wireless Mesh Networks. In *Proc. 3rd IEEE Conf. on Wireless On demand Network Syst. and Services (WONS'06)*, pages 59–67. IEEE Computer Society, 2006.

[37] R. Grosu and S. A. Smolka. Monte carlo model checking. In *Proc. 11th Int. Conf. on Tools and*

Algorithms for Construction and Analysis of Syst. (TACAS'05), volume 3440 of *LNCS*, pages 271–286. Springer, 2005.

[38] G. Guirado, T. Hérault, R. Lassaigne, and S. Peyronnet. Distribution, approximation and probabilistic model checking. *ENTCS*, 135(2):19–30, 2006.

[39] J. M. Hammersley and K. W. Morton. Poor Man's Monte Carlo. *J. Royal Statistical Soc. Series B (Methodological)*, 16(1):23–38, 1954.

[40] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, 1994.

[41] B. R. Haverkort. Are stochastic process algebras good for performance and dependability evaluation. In *Proc. Int. Workshop on Process Algebra and Performance Modeling*, volume 1853 of *LNCS*, pages 501–510. Springer, 2000.

[42] B. R. Haverkort. Markovian models for performance and dependability evaluation. In *European Educ. Forum: School on Formal Methods and Performance Analysis*, volume 2090 of *LNCS*, pages 38–83. Springer, 2002.

[43] B. R. Haverkort, H. Hermanns, and J.-P. Katoen. On the use of model checking techniques for dependability evaluation. In *Proc. 19th IEEE Symp. on Reliable Distributed Syst. (SRDS'00)*, pages 228–237. IEEE Computer Society, 2000.

[44] T. Hérault, R. Lassaigne, F. Magniette, and S. Peyronnet. Approximate probabilistic model checking. In *Proc. 5th Int. Conf. on Verification, Model Checking and Abstract Interpretation (VMCAI'04)*, volume 2937 of *LNCS*, pages 73–84. Springer, 2004.

[45] A. Hinton, M. Z. Kwiatkowska, G. Norman, and D. Parker. Prism: A tool for automatic verification of probabilistic systems. In *Proc. 2nd Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'06)*, volume 3920 of *LNCS*, pages 441–444. Springer, 2006.

[46] G. Holzmann. *The Spin Model Checker, Primer and Reference Manual.* Addison-Wesley, 2003.

[47] J. Hromkovic, R. Klasing, B. Monien, and R. Peine. Dissemination of information in interconnection networks (Broadcasting & Gossiping). *Combinatorial Network Theory*, pages 125–212, 1996.

[48] M. Huth and M. Ryan. *Logic in Computer Science: Modelling and reasoning about systems.* Cambridge University Press, Cambridge, UK, 2004.

[49] M. Jelasity, R. Guerraoui, A.-M. Kermarrec, and M. van Steen. The peer sampling service: Experimental evaluation of unstructured gossip-based implementations. In *Proc. 5th ACM/IFIP/USENIX Int. Middleware Conf. (Middleware'04)*, volume 3231 of *LNCS*, pages 79–98. Springer, 2004.

[50] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking. Randomized rumor spreading. In *Proc. 41st Annu. IEEE Symp. on Found. of Comput. Sci. (FOCS'00)*, pages 565–574. IEEE Computer Society, 2000.

[51] J.-P. Katoen, T. Kemna, I. Zapreev, and D. Jansen. Bisimulation minimisation mostly speeds up probabilistic model checking. In *Proc. 13th Int. Conf. on Tools and Algorithms for Construction and Analysis of Syst. (TACAS'07)*, volume 4424 of *LNCS*, pages 76–92. Springer, 2007.

[52] J.-P. Katoen, D. Klink, M. Leucker, and V. Wolf. Three-valued abstraction for continuous-time Markov chains. In *Proc. 19th Int. Conf. on Comput. Aided Verification (CAV'07)*, 2007. To appear.

[53] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Proc. 44th Annu. IEEE Symp. on Found. of Comput. Sci. (FOCS'03)*, pages 482–491. IEEE Computer Society, 2003.

[54] D. Kempe, J. Kleinberg, and A. Demers. Spatial gossip and resource location protocols. In *Proc. 33rd Annu. ACM Symp. on Theory of Computing (STOC'01)*, pages 163–172. ACM Press, 2001.

[55] D. Kempe and J. M. Kleinberg. Protocols and impossibility results for gossip-based communication mechanisms. In *Proc. 43rd Annu. IEEE Symp. on Found. of Comput. Sci. (FOCS'02)*, pages 471–480. IEEE Computer Society, 2002.

[56] M. Kwiatkowska, G. Norman, and D. Parker. Game-based abstraction for markov decision processes. In *Proc. 3rd Int. Conf. on the Quantitative Evaluation of Syst. (QEST'06)*, pages 157–166. IEEE Computer Society, 2006.

[57] S. Laplante, R. Lassaigne, F. Magniez, S. Peyronnet, and M. de Rougemont. Probabilistic abstraction for model checking: An approach based on property testing. In *Proc. 17th IEEE Symp. on Logic in Comput. Sci. (LICS 2002)*, pages 30–39. IEEE Computer Society, 2002.

[58] J. Luo, P. T. Eugster, and J.-P. Hubaux. Route driven gossip: Probabilistic reliable multicast in ad hoc networks. In *Proc. 22nd Conf. of the IEEE Comput. and Commun. Soc. (INFOCOM 2003)*, volume 3, pages 2229–2239. IEEE Computer Society, 2003.

[59] A. McIver and A. Fehnker. Formal techniques for the analysis of wireless networks. In *Proc. 2nd Int. Symp. on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA'06)*, 2006.

[60] D. Mosk-Aoyama and D. Shah. Computing separable functions via gossip. In *Proc. 25th Annu. ACM Symp. on Principles of Distributed Computing (PODC'06)*, pages 113–122. ACM Press, 2006.

[61] T. Nipkow, L. C. Paulson, and M. Wenzel. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS*. Springer, 2002.

[62] G. Norman, D. Parker, M. Z. Kwiatkowska, S. K. Shukla, and R. Gupta. Using probabilistic model checking for dynamic power management. *Formal Aspects of Computing*, 17(2):160–176, 2005.

[63] S. Owre, S. Rajan, J. M. Rushby, N. Shankar, and M. K. Srivas. PVS: Combining Specification, Proof Checking, and Model Checking. In *Proc. 8th Int. Conf. on Comput. Aided Verification (CAV'96)*, volume 1102 of *LNCS*, pages 411–414. Springer, 1996.

[64] J. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D. Epema, M.Reinders, M. van Steen, and H. Sips. Tribler: A social-based peer-to-peer system. *Concurrency and Computation:*

Practice and Experience, 19:1–11, 2007.

[65] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* John Wiley & Sons, Inc., New York, NY, USA, 1994.

[66] D. Randall. Rapidly Mixing Markov Chains with Applications in Computer Science and Physics. *Computing in Sci. and Eng.*, 8(2):30–41, 2006.

[67] P. Schnoebelen. The verification of probabilistic lossy channel systems. In *Validation of Stochastic Systems*, volume 2925 of *LNCS*, pages 445–466, 2004.

[68] K. Sen, M. Viswanathan, and G. Agha. Statistical model checking of black-box probabilistic systems. In *Proc. 16th Int. Conf. on Comput. Aided Verification (CAV'04)*, volume 3114 of *LNCS*, pages 202–215. Springer, 2004.

[69] R. van Renesse, K. P. Birman, and W. Vogels. Astrolabe: A robust and scalable technology for distributed system monitoring, management, and data mining. *ACM Trans. Comput. Syst.*, 21(2):164–206, 2003.

[70] R. van Renesse, Y. Minsky, and M. Hayden. A gossip-style failure detection service. In *Proc. 1st IFIP Conf. on Distributed Syst. Platforms and Open Distributed Processing (Middleware'98)*, pages 55–70. Springer, 1998.

[71] W. Vogels, R. van Renesse, and K. Birman. The power of epidemics: robust communication for large-scale distributed systems. *ACM SIGCOMM Comput. Commun. Review*, 33(1):131–135, 2003.

[72] S. Voulgaris. *Epidemic-Based Self-Organization in Peer-to-Peer Systems.* Doctoral thesis, Vrije Universiteit Amsterdam, 2006. Appears in Collections: Proefschriften Exacte Wetenschappen.

[73] S. Voulgaris, D. Gavidia, and M. van Steen. Cyclon: Inexpensive membership management for unstructured p2p overlays. *J. Network and Syst. Manage.*, 13(2):197–217, 2005.

[74] H. Younes, M. Kwiatkowska, G. Norman, and D. Parker. Numerical vs. statistical probabilistic model checking. *Software Tools for Technology Transfer (STTT)*, 8(3):216–228, 2006.

[75] H. L. S. Younes and R. G. Simmons. Probabilistic verification of discrete event systems using acceptance sampling. In *Proc. 14th Int. Conf. on Comput. Aided Verification (CAV'02)*, volume 2404 of *LNCS*, pages 223–235. Springer, 2002.