ISSN No. 0976-5697

Volume 2, No. 4, July-August 2011

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

Formal Verification of Ad Hoc Network Routing Protocols

Amandeep Verma
Assistant Professor in Computer Science,
Punjabi University Regional Centre for IT & Mgmt., Mohali
vaman71@gmail.com

Abstract: The conventional wireless network makes use of the base station, hardware, as a central entity. When no base station on hand or it is out of range, mobile nodes can still form a fully connected wireless network, accordingly called ad hoc mode. Ever since the initiative of ad hoc routing was regarded, an overabundance of protocols has surfaced, customized for a particular state. If a routing protocol does not achieve as projected, it is reduced quality of service. The routing protocol should be validated before deployment. A good way to confirm a protocol is to use formal verification techniques. This paper presents the various tools/techniques/languages used for formal modeling and verification of the ad hoc network routing protocols. It ranges from Petri nets, SPIN Model Checker, PROMELA, AVISPA, HLPSL, UPPAAL Model Checker, SDL and BAN logic.

Keywords: ad hoc networks, AVISPA, BAN Logic, formal verification, Petri nets, SDL, SPIN, UPPAAL

I. INTRODUCTION

The conventional wireless network makes use of the base station, hardware, as a central entity. For that reason, it is an infrastructure based network. The base station is coupled to a wired network and facilitates computers in proximity to hook up to it using wireless network cards. When no base station on hand or it is out of range, mobile nodes can still form a fully connected wireless network, accordingly called ad hoc mode. Nodes within wireless transmission range of each other can then communicate directly in a peer-to-peer fashion. A Mobile Ad hoc Network is a network of mobile nodes operating in ad hoc mode. In the case of an ad hoc network without multi-hop paths, a node that needs to deliver a packet to another node which is not within direct transmission range will not be able to do so.

This state can be coped by running a routing protocol all through the network so as to ascertain such paths. If there exists a multi-hop radio path involving two nodes, then packets can be routed. Building a routing protocol for a mobile ad hoc network poses added challenges in contrast to the infrastructure based case. Ever since the initiative of ad hoc routing was regarded, an overabundance of protocols has surfaced, customized for a particular state. If a routing protocol does not achieve as projected, it is reduced quality of service. In the most awful situation an application might not work by any means. Another quandary with a faulty protocol is that the workstation happens to be exposed to attacks from malicious users.

With the aim of to rule out invalid actions, protocol designers subject their designs to validation. A general manner to do this is to build a software model of the protocol and simulate a large figure of usage situations. The model is performed on virtual devices in a simulated environment. After a stipulated time the simulation is stopped and the result examined. A second methodology used in progress of the protocol is to do live tests on real hardware. In order for experimentations to give in significant information, they could do with a number of tests can be performed and averaged. Neither simulation nor testing is complete. They can be used to identify the bugs that are easily located but cannot exclude protocol blueprint errors. The third method to confirm a protocol is to use formal verification techniques. A protocol model is checked to see if it verifies to user requirements. Rather than executing the model, its logical formation is considered. This practice works by using mathematical logic and can be more or less automatic. It is wise to consider that formal verification is not a replacement for testing or simulation. These three techniques are much more complementary rather than competitive approaches.

There are mainly three kinds of automated formal verification techniques [8], namely, model checking, theorem proving and equivalence checking. Model checking is a method to validate if a formally modeled system satisfies a given property. Theorem proving technique uses mathematical methods, such as axioms and rules, to prove the correctness of a system. Equivalence checking formally checks if two models, at different abstraction levels, are equivalent.

Model Checking and Theorem Proving are used techniques for the validation of routing protocols of ad hoc networks. The literature has shown the various tools/models/ languages used for this purpose. This paper is intended for the audience, looking for the tools and techniques for formal modeling and verification of ad hoc network routing protocols. To determine the correctness of a particular protocol definition for correct operation of an ad hoc routing protocol [31] is like as follows. If there at one point in time exists a path between two nodes, then the protocol must be able to find some path between the nodes. When a path has been found, and for the time it stays valid, it shall be possible to send packets along the path from the source node to the destination node.

The rest of the paper is organized as follows. The section 2 is about the various techniques/tools/ languages used for the formal modeling and verification of ad hoc network routing protocols. The section 3 is the Conclusion.

II. REVIEW OF LITERATURE

There are number of formal approaches and their applications in diverse areas for validation and for proving correctness the models in study. The formal verification techniques applicable to all areas of ad hoc network namely, authentication, access control, routing etc. As the present study is about the usage of formal approach for the validation of routing protocols of ad hoc networks, so below is the listing of the techniques/tools for specification, modeling and verification of it with the references of studies that have used these for validation of their models.

A. Petri Nets:

Petri nets [21] are a basic model of parallel and distributed systems, designed by Carl Adam Petri in 1962. These are a graphical tool for the formal description of the flow of activities in complex systems. The technique is mathematically defined. Many static and dynamic properties of a Petri net (and hence a system specified using the technique) may be mathematically proven. With the time there many variants of Petri nets like High Level Petri nets, Fuzzy Petri nets, Object Oriented Petri nets, Place/Transition nets, Petri nets with time, Stochastic Petri nets, Hybrid, Modular, Inhibitor, Symmetric, Hierarchical and Colored Petri nets. The Petri net tools database for various variants and platforms is available on webpage [9].

Stochastic Petri Nets [36] is used to build an approximate model for a quick numerical analysis of performance. It allows the quick construction of **a** simplified abstract model that is numerically solved for different model parameters. The dynamic topological Fuzzy timing high level Petri nets [32] defined to construct and verify routing protocols for MANET. A highly abstract Coloured Petri Net model [4] of routing in a MANET based on DSDV routing protocol created. This study demonstrates the feasibility of using CPNs to model routing protocols of MANETs given their dynamically changing network topologies. An adequate formal modeling technique that is integration of Petri nets and Petri net transformations [14] called reconfigurable systems and algebraic higher order nets proposed for formal modeling and analysis of flexible processes in mobile ad-hoc networks

B. SPIN and PROMELA:

SPIN is a popular open-source software tool [11], used by thousands of people worldwide that can be used for the formal verification of distributed software systems. The tool was developed at Bell Labs in the original UNIX group of the Computing Sciences Research Center, starting in 1980. The software has been available freely since 1991, and continues to evolve to keep pace with new developments in the field. In April 2002 the tool was awarded the prestigious System Software Award for 2001 by the ACM. It has been used to detect design errors in applications ranging from high-level descriptions of distributed algorithms to detailed code for controlling telephone exchanges. SPIN verification models are focused on proving the correctness of process interactions, and they attempt to abstract as much as possible from internal sequential computations. SPIN accepts design specifications written in the verification language PROMELA (a Process Meta Language), and it accepts correctness claims specified in the syntax of standard Linear Temporal Logic (LTL).

PROMELA is a verification modeling language. It provides a way for making abstractions of distributed systems. It defines a finite set of processes, which together constitute the behavior of the system. Linear Temporal Logic a logic in which one express property of paths in a computation tree. In particular, properties such as “for some state on the path” or “for every two consecutive states” can be expressed.

A formal Verification [23] of Ad-Hoc Routing Protocols Using SPIN Model Checker is done. The protocol under study was WARP. In order to exemplify the methodology [5] for formal verification of routing protocols for ad hoc networks use PROMELA and verified them using SPIN. The verification of simplified version of CAR for verification [18] PROMELA and SPIN was used. A formal security analysis [12] of Secure AODV (SAODV) using SPIN studied. First, they formally specify two security properties in the presence of an external attacker and model the protocol using PROMELA, the specification and modeling language of SPIN. A Component based Testing Technique for a MANET Routing Protocol [35] use PROMELA and SPIN to build formal model for their study.

C. UPPAAL:

UPPAAL is a tool box for *validation* (via graphical simulation) and *verification* (via automatic model checking) of real-time systems. The tool has been developed in collaboration between the Design and Analysis of Real-Time Systems group at Uppsala University, Sweden and Basic Research in Computer Science at Aalborg University, Denmark. UPPAAL uses very restricted data structures, forcing to find alternative ways of expressing some complex data structures. This resulted in an increase in model size

A study to define methodology [26] for the verification of real-world communication protocols use UPPAAL to verify timing properties of AODV protocol and results a tractable timed automata model of AODV. For automatized verification [31] of Ad hoc Routing protocols the paper evaluates two model checking tools, SPIN and UPPAAL, using the verification of the Lightweight Underlay Network Ad hoc Routing protocol (LUNAR) as a case study. A pattern to model mobile ad hoc networks in UPPAAL, including encodings of locations and mobility as well as local broadcast where the actual receivers of messages are those nodes only that are immediate neighbours of the emitting node was provided [13].

D. AVISPA and HLPSL:

AVISPA stands for *Automated Validation of Internet Security Protocols and Applications*. It provides a modular and expressive formal language for specifying protocols and their security properties, and integrates different back-ends that implement a variety of automatic protocol analysis techniques [16]. SPAN, the Security Protocol ANimator for AVISPA is designed to help protocol developers in writing HLPSL specifications. From an HLPSL specification SPAN helps in interactively building Message Sequence Charts (MSC) of the protocol execution. Since SPAN implements an active intruder, it can also be used to interactively find and build attacks over protocols. The High Level Protocol Specification Language (HLPSL) is an expressive language [34] for modelling communication and security protocols. HLPSL draws its semantic roots from Lamport’s Temporal Logic of Actions. The importance of AVISPA is given by the fact that it has a high capacity of developing new network protocols and of securing already proposed protocols, making them easier to accept by the users.

The demonstration of AVISPA formal verification tool can be used to validate the security properties of ad hoc secure routing protocols is presented **[3][19][20]**. In order to prove the technique formal verification of ARAN, endairA are taken as case study.

E. SDL:

Specification and Description Language (SDL) is an object-oriented, formal language [24] defined by the International Telecommunications Union- Telecommuni cations Standardization Sector (ITU-T). The commercial tools available in the market are ObjectGeode, Cinderella, Safire-SDL, PragmaDev. The variant SDL-RT is used to develop real-time and embedded software.

In the validation model [6][7][30] of DSR Protocol formal specifications use SDL Language.

F. Specification language Z:

The Z (pronounced Zed) language is a formal specification language named after Zermelo–Fraenkel set theory, is a formal specification language used for describing and modeling computing systems. Z was developed in Paris, France and Oxford, England. **Z/EVES,** is a proof tool based on EVES and ZF set theory that supports the Z notation. The Z notation is used as a formal technique because of its abstract characteristics and properties, and having a rigorous computer tool support.

The Z notation is used as a formal technique [28] for formal Verification of Route Request procedure for AODV Protocol. The formal specification is analyzed and validated using Z Eves tool.

G. BAN Logic:

Burrows–Abadi–Needham logic (also known as the BAN logic) is a set of rules for defining and analyzing information exchange protocols. Specifically, BAN logic helps its users determine whether exchanged information is trustworthy, secured against eavesdropping, or both. BAN logic starts with the assumption that all information exchanges happen on media vulnerable to tampering and public monitoring.

The paper [22] and another [29] describe the formalization of routing protocols by using BAN logic. A Secure Dynamic Source Routing protocol for Mobile Ad hoc Networks that prevents a lot of potential attacks to these kind of networks. The stated security goals using the BAN logic formalism [10] are shown.

H. SMV:

SMV [25] is another model checking tool used for verification of hardware systems.. It allows several forms of specification, including the temporal logics CTL and LTL, finite automata, embedded assertions, and refinement specifications. It also includes an easy-to-use graphical user interface and source level debugging capabilities. SMV automatically verifies a design for all possible input sequences for properties of combinational logic and interacting finite state machines. When a property fails to verify, then a counterexample trace is produced which helps locating the bugs and fixing the model. NuSMV [1] is an open source tool for Symbolic Modeling Checking.

The most well-known tool that makes use of symbolic model checking is SMV series. In this paper we use Cadence SMV, which is designed in Cadence Berkeley Lab. The properties to be verified are specified in LTL (Linear Temporal Logic). Although originally it is designed mainly as a verification system for hardware design, many people started to apply it to more general and software-based verification tasks. The usage of Cadence SMV for validation of AODV protocol is shown in the study [33].

I. Others:

The paper [2] describes the modeling of AODV, a reactive routing protocol for MANETs, in the ω-calculus.

A technique for modeling and automatic verification of network protocols, based on graph transformation [17], is suggested. It is suitable for protocols with a potentially unbounded number of nodes, in which the structure and topology of the network is a central aspect, such as routing protocols for ad hoc networks.

III. CONCLUSION

Formal verification is a promising technique to validate algorithms for wireless networks. It presents an expressive increase in the quality of the protocol. The practices presented here are a fine beginning for people who would like to pursue the research on this field or apply formal verification on their algorithms. Theorem proving mechanisms involves manual interaction and on the other hand, model checking is almost automatic. It is difficult to say that one approach is the replacement of some other approach because each has merits and demerits. The trade off is among the complexity, accuracy and automaton.

IV. REFERENCES

[1] A. Cimatti, E. M. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, M. Roveri, R. Sebastiani and A. Tacchella, “NuSMV 2: An OpenSource Tool for Symbolic Model Checking” In Proceeding of International Conference on Computer-Aided Verification, Copenhagen, Denmark, July 27-31, 2002

[2] Anu Singh, C. R. Ramakrishnan, and Scott A. Smolka, “Modelng the Aodv Routing Protocol in the ω-CALCULUS”, Proceedings of IEEE Conference on System, Applications and Technology, 2006, pp. 1-5

[3] Benetti Davide , Merro Massimo and Vigan Luca, “Model Checking Ad Hoc Network Routing Protocols: ARAN vs. endairA”, Proceedings of IEEE International Conference on Software Engineering and Formal Methods, Sept. 2010, pp. 191 - 202

[4] C. Yuan, and J. Billington, “An Abstract Model of Routing in Mobile Ad Hoc Networks”, Sixth Workshop and Tutorial on Practical Use of CPN and the CPN Tools, October 2005, pp. 137 – 156.

[5] Camara D, Loureiro A. A. F. and Filali F, “Methodology for Formal Verification of Routing Protocols for Ad Hoc Wireless Networks”,Proceedings of the IEEE Conference on Global Communications, November 2007, pp. 705 - 709

[6] Cavalli, A., Grepet, C., Maag, S. and Tortajada, V., “A validation model for the DSR protocol”, Proceedings of the International Conference on Distributed Computing, 2004, pp. 768-773

[7] Cyril Grepet and Stephane Maag, “A Testing Methodology for a MANET Routing Protocol using a Node Self-Similarity Approach”, Proceedings of the Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops, 2007, pp. 514 – 519.

[8] Daniel Câmara, Antonio A.F. Loureiro and Fethi Filali, , “Formal Verification of Routing Protocols for Wireless Ad Hoc Networks”, Guide to Wireless Adhoc Networks DOI: 10.1007/978-1-84800-328-6_8, 2009, pp. 189 - 210

[9] Frank Heitmann and Daniel Moldt, http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/db.html

[10] Frank Kargl, Alfred Geis, Stefan Schlott, Michael Weber, “Secure Dynamic Source Routing”, Proceedings of the 38th Annual Hawaii International Conference on System Sciences, 2005, pp. 320c-320c

[11] Gerard J. Holzmann, “The Model Checker SPIN”, IEEE Transactions On Software Engineering, 1997, Vol. 23, No. 5, pp. 1-17.

[12] Gurdag A. Burak and Caglayan M. Ufuk, “A Formal Security Analysis of Secure AODV (SAODV) using Model Checking”. Proceedings of the International Symposium on Computer networks, June 2008

[13] Jens Chr, Godskesen and Olena Gryn, “Modelling and verification of security protocols for ad hoc networks using UPPAAL”, Proceedings of the 18th Nordic Workshop on Programming Theory (NWPT'06), October 2006

[14] Kathrin Hoffmann, “Formal Modeling and Analysis of Mobile Ad Hoc Networks and Communication Based Systems using Graph and Net Technologies”, Bulletin of the European Association for Theoretical Computer Science, 2010, No. 101, pp. 148-160

[15] Levente Butty´an and Ta Vinh Thong, “Formal verification of secure ad-hoc network routing protocols using deductive model-checking”, Proceedings of Wireless and Mobile Networking Conference, December 2010, pp. 1-6..

[16] Luca Vigan, “Automated Security Protocol Analysis With the AVISPA Tool”, Electronic Notes in Theoretical Computer Science, 2006, pp. 61-86.

[17] Mayank Saksena, Oskar Wibling, and Bengt Jonsson, “Graph Grammar Modeling and Verification of Ad Hoc Routing Protocols”, Proceedings of the 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), Vol. 4963 of Lecture Notes in Computer Science, Springer Verlag, March 2008, pp 18–32.

[18] Maysam Yabandeh, Reza Shokri, and Nasser Yazdani, “Formal Verification of CAR for Wireless Ad Hoc Networks”, Proceedings of the International Symposium on Software Engineering, 2007, pp. 462 – 465

[19] Mihai-Lica Pura, Victor-Valeriu Patriciu, Ion Bica, “Formal verification of secure ad hoc routing protocols using AVISPA: ARAN case study”, Proceedings of the 4th conference on European computing conference ECC'10, ISBN: 978-960-474-178-6, pp. 200 - 206

[20] Mihai-Lica Pura, Ion Bica and Victor-Valeriu Patriciu, , “On Modeling and Formally Verifying Secure Explicit On-Demand Ad Hoc Routing Protocols”, Proceedings of International Conference on Software Engineering and Technology, October 2010, Vol. 2, pp. 215 - 220

[21] Murata, T., “Petri nets: Properties, analysis and applications”, Proceedings of the IEEE, 1989, Vol. 77, Issue 4, pp. 541-580.

[22] Qiuna Niu, “Formal Analysis of Secure Routing Protocol for Ad Hoc Networks”, Proceedings of International Conference on Wireless Communication and Signal Processing, November 2009, pp. 1-4

[23] Renesse, F. and Aghvami, A.H., “Formal verification of ad-hoc routing protocols using SPIN model checker”, Proccedings of the IEEE Mediterranean Electrotechnical Conference, 2004, Vol 3, pp. 1177 – 1182.

[24] Rockstrom, A. and Saracco, R., “SDL--CCITT Specification and Description Language”, IEEE Transactions on Communications, Vol. 30, Issue 6, June 1982, pp. 1310 – 1318.

[25] SMV language reference, http://www.cs.wpi.edu/kfisler/Courses/525V/S02/Readings/smv-cadence.pdf

[26] S. Chiyangwa and M. Kwiatkowska, “Modeling Ad hoc On-demand Distance Vector (AODV) Protocol with Timed Automata”, Proceedings of Third Workshop on Automated verification of Critical Systems (AVoCS'03), Southampton April 2003

[27] Shakeel Ahmed, A. K. Ramani and Nazir Ahmad Zafar, “Verifying Route Request Procedure Of AODV Using Graph Theory And Formal Methods” International Journal on Applications of Graph Theory in Wireless Ad hoc Networks and Sensor Networks , 2011, Vol. 3, Issue 2, pp. 1-13

[28] Shakeel Ahmed,A. K. Ramani and Nazir Ahmad Zafar, “Formal Verification of Route Request Procedure for AODV Routing Protocol”, International Journal of Advanced Research in Computer Science, 2011, Vol. 2, No. 1, pp. 432-536.

[29] Shu-Dong Shi, “A Formal Verification and Improvement of a Secure Protocol for Adhoc Networks”, Proceedings of International conference on Wireless Communications, Networking and Mobile Computing, 2008, pp. 1-3

[30] Stéphane Maag and Cyril Grepet and Ana Cavalli, “A formal validation methodology for MANET routing protocols based on nodes' self similarity”, Journal of Computer Communications, March 2008, Vol. 31, No. 4, pp. 827-841.

[31] Wibling, O., Parrow, J., and Pears, A.N., “Automatized Verification of Ad Hoc Routing Protocols”, In Proceedings of International Conference on Formal Techniques for Networked and Distributed Systems FORTE, 2004, Vol. 3235 of Lecture Notes in Computer Science, Springer Verlag, September 2004, pp. 343–358.

[32] Xiong Chaotue, Murata Tadao and Leigh Jason, “An Approach for verifying Protocols in Mobile Ad hoc Networks using Petri Nets”, Proceedings of the IEEE Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communications, 2004, Vol 2, pp. 537-540.

[33] Xin Liu and Jun Wang, “Formal Verification of Ad hoc On-demand Distance Vector (AODV) Protocol using Cadence SMV”, CPSC513 course project Report, Univ. of British Columbia, 2004

[34] Yannick Chevalier, Luca Compagna, Jorge Cuellar, Paul Hankes Drielsma, Jacopo Mantovani, Sebastian M¨odersheim, and Laurent Vigneron, “A High Level Protocol Specification Language for Industrial Security-Sensitive Protocols”, Volume 180 of Automated Software Engineering,. Austrian Computer Society, Austria, September 2000, pp 193–205

[35] Zaidi, F., Lallali, M. and Maag, S. , “A Component based Testing Technique for a MANET Routing Protocol”, Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, 2010, pp. 1-7

[36] Zhang C. and Zhou M., “A stochastic Petri net-approach to modeling and analysis of ad hoc network”, Proceedings of International Conference on Information Technology, 2003, pp. 152-156.