Automatic Functional Verification of OPNET Models with SDL-OPNET Co-Simulation

Tae-Hyong Kim, Qi-Ping Yang, Jae-Woo Kim

Kumoh National Institute of Technology, Gumi, Korea

Summary
Owing to the model-driven development (MDD) technology, reliable network protocols could be developed with an integrated and systematic way. As performance, another core metric for evaluation of network protocols, is usually not the target of MDD process, how to connect performance evaluation to the MDD process is an important issue. This paper presents a method to enhance the reliability of performance evaluation when a performance model of a network protocol should be designed separately in a different performance simulation tool. The proposed method verifies the functional correctness of the OPNET model of a network protocol generated from its original SDL model designed with an SDL-based MDD process by SDL-OPNET co-simulation. A test system designed in SDL is used to automate the verification process and all the components of the verification system could be obtained systematically in the proposed method. Experimental results with the logical link control (LLC) protocol show the applicability of the proposed method. The proposed method could be also applied to other MDD and performance simulation tools only if they provide external interfaces for co-simulation.

Key words:
Model-driven development, model verification, Performance evaluation, Co-simulation

1. Introduction

As the requirements of network protocols are getting more various and complicated, development of correct and reliable protocol implementations has been a major issue in protocol engineering. Conformance testing is a standard method to obtain that goal by checking if a protocol implementation conforms to the standard or the specification of that protocol [1]. The formal description techniques such as the specification and description language (SDL) [2] enabled formal methods in conformance testing and automatic protocol development with model-driven approaches [3-5]. Several powerful model-driven development (MDD) tools such as IBM's Rational Tau [6] are currently in use to design, verify, and implement network protocols.

Performance evaluation is also necessary in designing a network protocol since performance is another core requirement of a network protocol. Among various performance evaluation techniques, simulation techniques with powerful performance simulation tools have been widely accepted as a practical and reliable method as network environments and protocol behaviors are getting more complicated. OPNET Modeler is a leading network performance simulation tool; it provides a lot of network models including latest wireless protocols and is used by numerous world-wide users in research and development areas [7]. Note that performance simulation tools use their own techniques in designing network models which are not compatible each other. Nor do they allow importing models designed by MDD technologies. In this situation, a protocol must be modeled repeatedly and separately for functional verification and for performance evaluation, which may result in inconsistency between the two models of a protocol. There have been several approaches to overcome this problem that try to integrate both functional verification and performance evaluation. Some studies developed new tools that support MDD with standard modeling languages and performance evaluation with some extensions on those languages [8-10], of which the universality would be restricted due to the specific tools and extensions on the standard. Others tried to use well-known MDD tools and performance simulation tools both by tool coupling or model mapping [11-13]. While modeling mapping, trying to automate performance model generation, has an advantage of individual use of tools, mapping process cannot be completely automated and the functional equivalence between two models cannot be guaranteed.

Recently we presented mapping rules from an SDL model of a network protocol to its corresponding OPNET model because two models have similar extended finite state machine (EFSM) structure and Tau generates executable C code from SDL models [13]. Our tool supports semi-automatic model conversion only and there is still a possibility of functional inconsistency between the models due to some manual conversion. The motivation of this work is to verify the functional conformance of a generated OPNET models to its original SDL model. There are several different approaches of simulation model verification as verification of simulation models is an important issue in modeling and simulation (M&S)

Manuscript received May 5, 2011
Manuscript revised May 20, 2011

systems. Most verification work of a performance simulation model, however, focuses on the performance of the model and usually uses manual process. This paper presents a method to verify the functionality of an OPNET model automatically when an OPNET model was generated from an SDL model designed by an MDD approach.

This paper is organized as follows. Section 2 introduces some related work on model verification of M&S systems and the proposed verification technique is explained in section 3. Section 4 shows an experimental result with a simple logical link control (LLC) protocol [14]. Finally conclusions are drawn in section 5.

2. Related Work

This section introduces two different verification approaches of performance simulation models and related case studies of verification of OPNET models as related work.

The first verification approach of performance simulation model majorly focuses on the performance correctness of the models. It usually compares simulation results with the reference data or experimental results with real network products. When the National Institute of Standards and Technology (NIST) developed an OPNET model for the dynamic source routing (DSR) protocol for mobile and ad-hoc networks (MANETs), they compared the simulation results with other published data [15]. B. Van den Broeck *et al.* tried to verify router models in OPNET by comparing with actual behaviors of real routers [16]. Nicola Baldo *et al.* validated the IEEE 802.11 medium access (MAC) layer model in ns-3 [17] using their EXTREME tested [18]. By this approach, performance models could be freed of some bugs and improved through calibration processes.

Functional correctness is the target of the other verification approach. A simple way to functional verification is to test the required functionalities of the model one by one manually. Developers of performance simulators normally issue verification reports of their models regularly which have verdict tables of each functional requirement. For complete verification of the model, formal methods could be used such as model checking. With the tools supporting those formal methods, this verification approach might be also automated. Karthikeyan Bhargavan *et al.* developed Verisim by connecting two existing tools, ns-2 [19] and the MaC monitoring and checking framework [20], which verifies some properties on traces produced by ns-2 with MaC's model checking feature [21]. Mamadou K. Traoré tried to perform theorem proving-based verification of the discrete event system specification (DEVS) models by specifying formal semantics on those models [22]. Ahmed Sobeih *et al.* proposed modified J-SIM, a Java-based simulator, to support state space exploration and verified the ad-hoc on-demand distance vector (AODV) routing protocol by model checking [23]. There has been no functional verification work of performance simulation models with MDD-based approach yet, where the verification process can be supported by MDD tools.

3. The Proposed Verification Technique

The goal of this paper is to automate the checking if the OPNET model of a network protocol generated from SDL-OPNET model mapping conforms to that original SDL model within the same MDD process. To obtain that goal we propose test architecture of an OPNET model with SDL-OPNET co-simulation techniques.

3.1 MDD-based Verification Approach

The proposed approach considers an efficient performance evaluation method of MDD models and tries to use a well-known powerful performance simulation tool. Fig. 1 shows how model mapping between formal models and performance simulation models is used in the general MDD process. While the formal model and the implementation are verified according to the specification or the standard for developing a reliable implementation, the performance simulation model converted form the formal model should be verified with respect to that formal model or the implementation for developing a performance-guaranteed implementation.

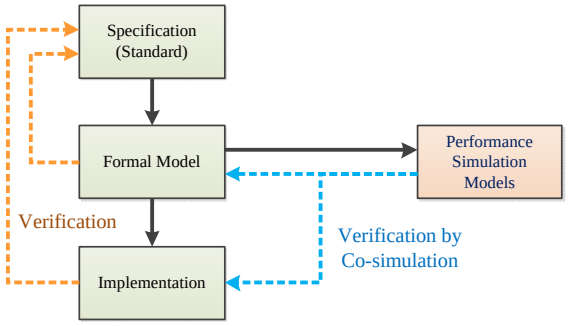

Fig. 1 The proposed MDD-based verification of performance simulation model

As the reference of performance simulation model verification is the original formal model, we use the co-simulation technique between the MDD tool and the performance simulation tool. With test harness for co-simulation designed in the MDD technique, verification process could be managed by the MDD tool in an integrated and automated method. IBM Rational Tau and OPNET Modeler are used in our verification method but any MDD tools and performance simulation tools could be

used also in this approach if both tools provide some external interfaces for co-simulation. Details of the proposed verification technique with co-simulation are shown in subsection 3.3.

3.2 SDL-OPNET Co-simulation

Co-simulation between IBM Rational Tau and OPNET Modeler can be realized with Tau's environment interface and Modeler's external system interface (ESI). Tau provides some input and output (I/O) functions for the environment, e.g., xInEnv() and xOutEnv(), in target code generation, which can be used for message exchanges with external code. Modeler supports co-simulation with other simulator by providing a special co-simulation package which includes external system definition (ESD), Esys kernel procedure, and external system access (ESA) application program interfaces (API). Fig. 2 shows the SDL-OPNET co-simulation structure of the proposed technique.

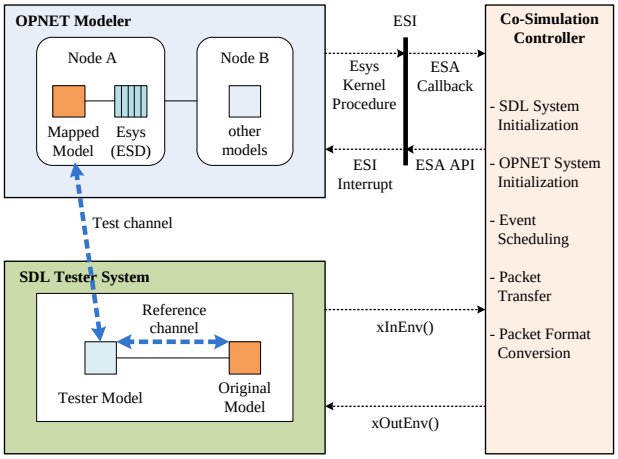

Fig. 2 Our SDL-OPNET co-simulation structure

The target model in Modeler exchanges messages with the test model designed in Tau through Modeler's ESI and Tau's environment functions. An external co-simulation controller was developed for realization of co-simulation; it initializes SDL and OPNET systems, handles event scheduling, and transfers packets by converting the packet format between SDL and OPNET models. Messages from the target model in OPNET to the tester model are sent by the matched Esys module which defines ESI special gateway. When the Esys module calls an Esys kernel procedure, the corresponding ESA callback function is called which was configured in ESI. Then, the callback function designed in the co-simulation controller sends the message to the tester model in Tau through xOutEnv() function after message format conversion. When the co-simulation controller receives a message from the tester model in Tau through xInEnv() function, it sends the message to ESI with ESA API after message format conversion. Then an ESI interrupt occurs and the Esys module can fetch the message with the Esys kernel procedure.

3.3 SDL Tester System

Verification of the target OPNET model is controlled and managed by an SDL tester system designed with Tau. The SDL tester system is composed of test I/O blocks, a test control block, and an optional reference block logically. For automatic and systematic verification, test cases can be generated by analyzing the original SDL model with structural model-based test generation methods such as the transition tour or the unique input output (UIO) method. In this approach, generated test cases are stored in the test control block with the test verdict process that sends a test message to OPNET Modeler through SDL-OPNET co-simulation and verifies the functional correctness of the target OPNET model by observing and analyzing incoming messages from OPNET Modeler. Instead of an individual design of test control block of each target protocol, general test control block could be used with a test reference block which contains the original SDL model. In this approach, the test control block sends messages to OPNET modeler and test reference block and performs on-the-fly analysis by comparing incoming messages from those two interfaces. This approach can be applicable to complete verification of the target model and would increase the portability of the test harness and the level of test automation.

Fig. 3 shows two node-based test architecture which can be used for simple verification with the inter-operability testing concept. The test I/O blocks are logically located on top of the target OPNET model and its peer and messages between the target model and the upper layer protocol are observed and controlled by the test control block.

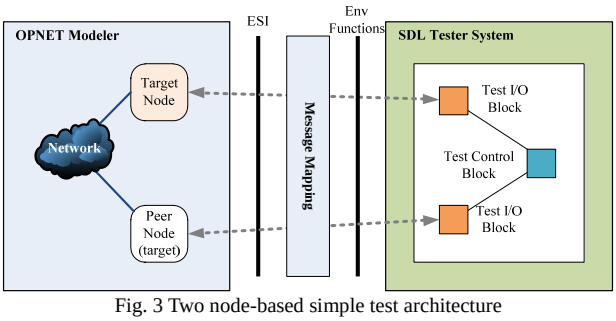

Fig. 3 Two node-based simple test architecture

More sophisticated test architecture is shown in Fig. 4 for complete verification of the target OPNET model. In addition to upper co-simulation interfaces of the target model, a lower interface is added for observing messages from the target model to the lower layer protocol. This

architecture is good for the general test control block with a reference block.

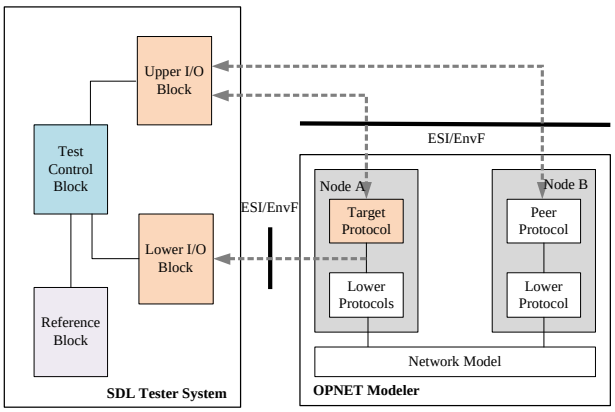

Fig. 4 Test architecture for complete functional verification

4. Experimental Results

For checking the efficacy of the proposed performance model verification technique, we designed a simple LLC protocol in SDL that supports go-back-N automatic repeat request (ARQ) [14]. With our model conversion tool that generates, from Tau generated C code of a SDL model, a corresponding OPNET model in a semi-automatic way, the target performance OPNET model was produced [13]. Then the functional verification system of the target model was constructed by designing an SDL-OPNET co-simulation controller and an SDL test system as shown in the previous section.

4.1 LLC model in SDL and OPNET

The top-level diagram of our simplified LLC system in SDL is shown in Fig.5. Two blocks, 'LLC_Sender' and 'LLC_Responder' for the required LLC functionalities use several primitive messages and protocol data units (PDU) for upper and lower layer interactions respectively which were defined according to the IEEE 802.2 standard. Blocks 'LLC_Tester' and "MAC' are added in the system for verification of 'LLC_Sender' and 'LLC_Responder' in by Tau's Simulator. Both the SDL processes in 'LLC_Sender' and 'LLC_Responder' have four states, 'Connected', 'Wait_Connected', 'Disconnected', and 'Wait_Disconnected'. Fig. 6 shows the extended finite state machine (EFSM) diagram of the converted LLC sender process model in OPNET modeler. Note that four SDL states have been mapped to the corresponding unforced states in red and two new states were added for handling SDL conditional transitions in the OPNET model.

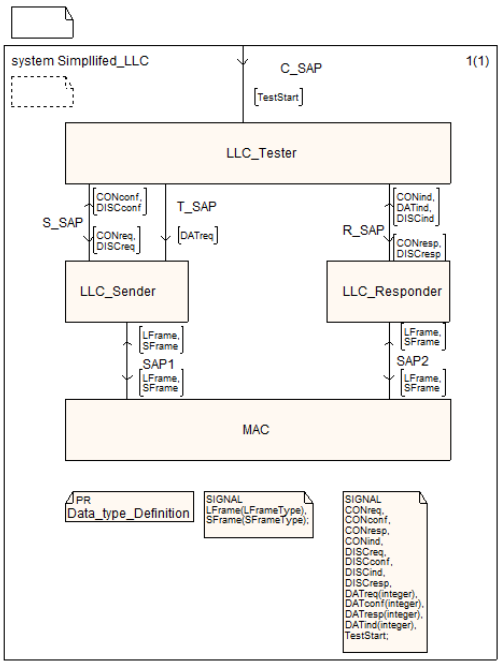

Fig. 5 SDL diagram of the simplified LLC system

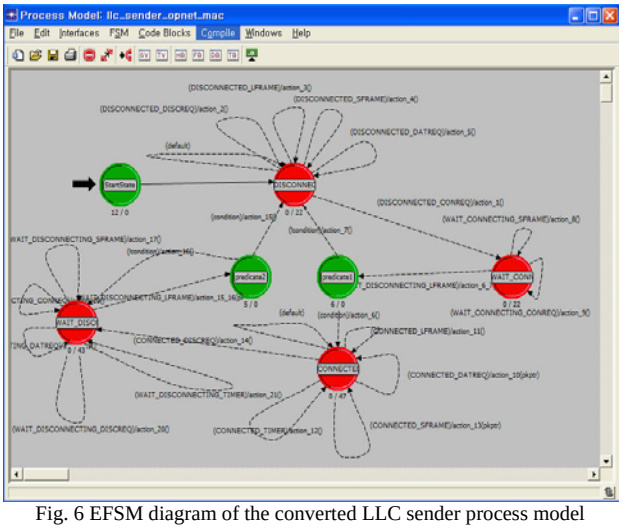

Fig. 6 EFSM diagram of the converted LLC sender process model

4.2 Co-simulation Controller and SDL Test System Design

For co-simulation between the target OPNET model and the SDL test system, co-simulation controller was developed in external C code which contains packet conversion between OPNET and SDL message formats. Fig.7 shows the node model diagram of OPNET's LLC sender where queue module 'llc_sender' is the target conversion model for verification and Esys module 'llc_socosim_test' was added for using the ESI defined in the ESD.

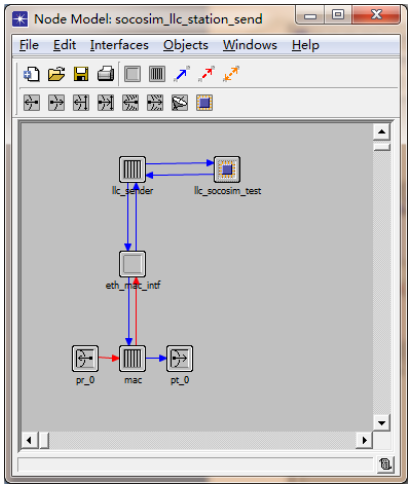

Fig. 7 EFSM diagram of the converted LLC sender process model

In this experiment, we used the two node-based test architecture shown in Fig.3 for its simplicity. The tester process of the test control block in the SDL test system is shown in Fig.8. This process sends test messages to the OPNET models and compares incoming messages through the co-simulation interface with the expected messages represented in the test cases generated separately.

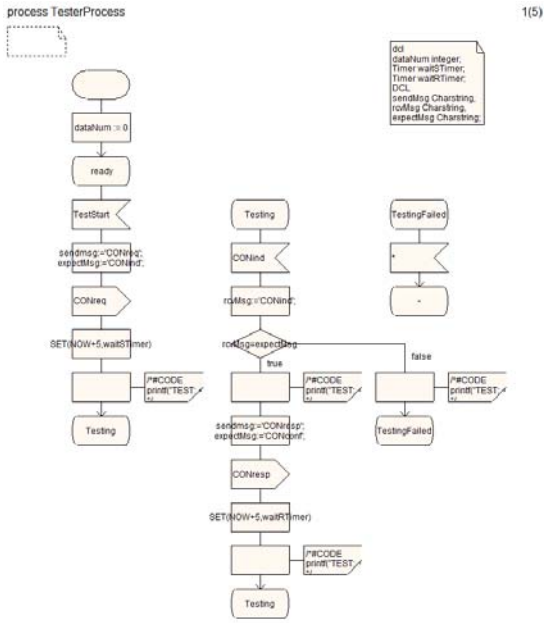

Fig. 8 EFSM diagram of the converted LLC sender process model

4.3 Verification Results

For functional verification of the LLC sender and receiver models in OPNET converted from the SDL models, we used an interoperability testing-level of test coverage because it can test the sender and receiver models at the same time. We generated 8 test cases the by transition tour method and implemented test verdict logic with those test cases in the test process of the SDL test system. All the system including the SDL test system, OPNET simulation models, and the external co-simulation controller system were compiled and linked to produce the executable code using the OPNET console, which is necessary if we use the co-simulation package provided by OPNET.

Fig.9 shows the verification process of the target OPNET model with the generated code. All textual messages except the boxed messages were produced by the SDL test system. Seven tests were passed among total eight tests. One test which sends the disconnecting request messages has not been passed because there was some incomplete part in the disconnection phase of the OPNET's LLC responder model. The SDL test system found that incompleteness of the converted target OPNET model, which was also reported by OPNET console debugger in boxed messages.

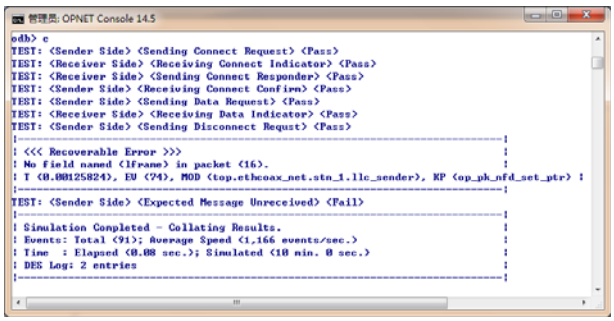

Fig. 9 Screenshot of the verification process with co-simulation

5. Conclusions

The MDD technology is an integrated and systematical method to obtain reliable products and MDD processes can be also easily automated. Performance requirements are also necessary points to be satisfied in developing network products. How to connect performance evaluation to the MDD process is therefore an important issue in the development of network protocols.

For the purpose of that goal, we tried to use existing well-known tools the reliability and competency of which have been proved in the market over time. Systematic coupling or linking different tools having peculiar design approaches is, however, very hard work. The problem that this paper attacked is how to guarantee the functional correctness of the OPNET models that were generated from the SDL model designed with Tau for reliable performance evaluation of that model. This paper presented an automatic functional verification method of such OPNET models with SDL-OPNET co-simulation.

The proposed performance model verification technique requires developing a co-simulation controller with appropriate message exchange and packet conversion between two models in SDL and OPNET and modifying the OPNET simulation model slightly with Esys module and ESD definition for co-simulation between Tau and Modeler. This work could be done straightly according to the design rule provided by the OPNET's co-simulation package with little possibility of critical faults affecting the verification results. The SDL tester system is the test harness which is necessary for the verification. Designing that system could be also systemized and even automated with analysis of the original SDL models. The proposed technique could be also applied to other MDD and performance simulation tools if they provide external interfaces for co-simulation.

We are planning to apply the proposed technique to more complicated network protocol such as resource management protocols of the long term evolution (LTE) system [24] to check its applicability. We are also interested in automatic generation of the test system that uses the testing and test control notation (TTCN) as well as SDL for better manageability of the verification process.

Acknowledgment

This paper was supported by Sabbatical Research Fund, Kumoh National Institute of Technology.

References

[1] ISO, "OSI Conformance Testing Methodology and Framework", IS-9646, 1991

[2] ITU, "Specification and Description Language", ITU-T Recommendation Z.100, 2000

[3] ITU-T, "Framework on formal methods in conformance testing", Recommendation Z.500, May 1997.

[4] Philipp Becker, Dennis Christmann, and Reinhard Gotzhein, "Model-driven development of time-critical protocols with SDL-MDD", In Proceedings of the 14th international SDL conference on Design for motes and mobiles (SDL'09), pp.34-52. 2009.

[5] Anas Showk, David Szczesny, Shadi Traboulsi, Irv Badr, Elizabeth Gonzalez, and Attila Bilgic. "Modeling LTE protocol for mobile terminals using a formal description technique", In Proceedings of the 14th international SDL conference on Design for motes and mobiles (SDL'09), pp.222-238, 2009.

[6] IBM Co. Ltd., Rational TAU SDL Suite, Ver. 6.3, 2009. See http://www-01.ibm.com/software/awdtools/tau/.

[7] OPNET Technology Inc., OPNET Modeler. See http://www.opnet.com.

[8] J. Hintelmann M. Diefenbruch and B. Muller-Clostermann, "Quest: Performance evaluation of sdl systems", In IFIP TC6/6.1, International Conference on Formal Description Techniques IX / Protocol Specification, Testing and Verification XVI, volume 69, pages 229–-244. Kluwer, 1996.

[9] C. U. Smith and L. G. Williams, "Performance Engineering Evaluation of Object Oriented Systems with SPE.ED", Int'l Conf. on Computer Performance Evaluation: Modeling Techniques and Tools, LNCS 1245, Springer, 1997.

[10] C. Schaffer, R. Raschholfer, and A. Simma, "EaSy-Sim: A Tool Environment for the Design of Complex, Real-Time Systems", Proc. Int'l Conf. on Computer Aided System Technologies, Springer, 1995.

[11] T. Kuhn, A. Geraldy, R. Gotzhein, and F. Rothländer, "ns+SDL - The Network Simulator for SDL Systems", SDL 2005 - Model Driven, Lecture Notes in Computer Science 3530, pp. 103-116, Springer, 2005.

[12] Qi-Ping Yang and Tae-Hyong Kim, "SDL-OPNET Co-Simulation Technique for the Development of Communication Protocols with an Integrated Approach to Functional Verification and Performance Evaluation", Journal of the Korea Society for Simulation, Vol.19, No.2, 2010.

[13] Jae-Woo Kim and Tae-Hyong Kim, "SDL-OPNET Model Conversion Technique for the Development of Communication Protocols with an Integrated Model Design Approach", Journal of the Institute of Embedded Engineering of Korea, Vol.5, No.2, 2010.

[14] IEEE, "Logical Link Control", IEEE Standard 802.2.

[15] J. Ballah. "OPNET DSR Verification and Validation". Thesis, Air Force Institute of Technology, Air University, March 2002.

[16] B. Van den Broeck, P. Leys, J. Potemans, J. Theunis, E. Van Lil, A. Van de Capelle, "Validation of Router Models in OPNET", OPNETWORK 2002.

[17] The ns-3 Network Simulator, See http://www.nsnam.org//.

[18] Nicola Baldo, Manuel Requena-Esteso, Jose Nunez-Martinez, Marc Portoles-Comeras, Jaume Nin-Guerrero, Paolo Dini, and Josep Mangues-Bafalluy. "Validation of the IEEE 802.11 MAC model in the ns3 simulator using the EXTREME testbed". The 3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools '10). 2010.

[19] The Network Simulator ns 2, http://www.isi.edu/nsnam/ns/.

[20] M. Kim, M. Viswanathan, H. Ben-Abdallah, S. Kannan, I. Lee, and O. Sokolsky. "Formally specified monitoring of temporal properties", In Proc. of ECRTS, 1999.

[21] K. Bhargavan, C. A. Gunter, M. Kim, I. Lee, D. Obradovic, O. Sokolsky, and M. Viswanathan, "Verisim: Formal analysis of network simulations", IEEE Transactions on Software Engineering, 28(2):129-145, February 2002.

[22] M. K. Traore, "Analyzing static and temporal properties of simulation models", In Proc. of the 2006 Winter Simulation Conference, 2006.

[23] Ahmed Sobeih, Mahesh Viswanathan, Darko Marinov and Jennifer C. Hou, "J-Sim: An Integrated Environment for Simulation and Model Checking of Network Protocols," Proceedings of the IEEE International Symposium on Parallel and Distributed Processing (IEEE IPDPS 2007), NSF Next Generation Software Program Workshop, Long Beach, CA, March 2007.

[24] 3GPP, "UTRA-UTRAN Long Term Evolution (LTE) and 3GPP System Architecture Evolution (SAE)", See http://www.3gpp.org/Highlights/LTE/LTE.htm.

Tae-Hyong Kim received the B.S. and M.S. degrees, from Yonsei University in 1992 and 1995, respectively, and a Ph.D. degree in electrical and electronic engineering from the same university in 2001. He was a postdoctoral fellow at the School of Information Technology and Engineering (SITE) at the University of Ottawa from 2001 to 2002. He is currently an assistant professor in the School of Computer and Software Engineering (SCSE) at the Kumoh National Institute of Technology (KIT) in Korea. His current research interests include software and protocol specification, verification and testing techniques, communication protocols, and next generation mobile networks. He is a member of the SDL Forum Society.

Qi-Ping Yang received the B.S. degree in communication engineering from Jilin University, Changchun, China, in 2001, and the M.S. degree in computer engineering from Kumoh National Institute of Technology (KIT), Gumi, Korea, in 2006. He is currently a Ph.D. candidate in the School of Computer and Software Engineering at KIT. His current research interest is software testing techniques and next generation mobile networks.

Jae-Woo Kim received the B.S. degree in computer and software engineering from Kumoh National Institute of Technology (KIT), Gumi, Korea, in 2004, and the M.S. degree in computer engineering from the same university in 2006. He is currently a Ph.D. candidate in the School of Computer and Software Engineering at KIT. His current research interest is protocol engineering and next generation mobile networks.