

A Programmable Interactive Approach for QualNet-based Network Model Evaluation and Testing

Haiqiao Wu^{**}, Lining Han^{**}, Li Huang^{*}, Meng Song^{**}, Duk Kyung Kim^{***}, and Peng Gong^{**}

^{*}State Grid Global Energy Interconnection Research Institute, Nanjing, China

^{**}National Key Laboratory of Mechatronic Engineering and Control, School of Mechatronic Engineering, Beijing Institute of Technology, Beijing, China

^{***}Department of Information and Communication Engineering, Inha University, Incheon, Korea
whqjoe@gmail.com, lininghan@bit.edu.cn, huangli@geiri.sgcc.com.cn, kdk@inha.ac.kr, penggong@bit.edu.cn

Abstract—In the previous work, an interactive real-time user interface (RTUI), which is a reliable human-in-the-loop framework that allows human users to dynamically modify the parameters of network model during the simulation process, is investigated. However, the delay overhead of parameter re-configuration commands transmission is unavoidable, especially for a great amount of parameters in a large scale network model. In this paper, a programmable dynamic simulation interface (PDSI) is further proposed. With the proposed PDSI, each of the pre-programmed re-configuration commands is marked with a time label, and then all the commands are encapsulated as packets and sent to the simulator in advance. Depending on the time label, the re-configuration command is pre-cached in the event queue and precisely implemented according to the pre-programmed order, which eliminates the delay overhead of command transmission for the parameter changing of network model. Simulation results demonstrate the efficiency of the proposed PDSI compared to the RTUI.

Keyword—PDSI, RTUI, pre-cache, programmable, QualNet

I. INTRODUCTION

Network simulation is a significant method to examine the effectiveness of network models at design and development process before implemented in real world [2-4].

Besides, the node placement strategies in wireless sensor networks can be evaluated effectively [5]. Compared to physical test, the simulation test method is more cost-effective and flexible, particularly for large-scale networks [6]. However, static network simulation (the parameters cannot be changed by human during the simulation process) causes many repetitive works during the process of evaluating and testing the impacts of parameters on network. In addition, many real-time events, such as node failures, are not considered during the simulation process in the static simulation method [7]. Although an on-line simulation framework for adaptive large-scale network parameter configuration [8] has been proposed to solve a variety of network protocols and configuration problems, the proposed solution cannot realize large-scale real-time simulations because of the delay caused by the recursive random search algorithm. An interactive real-time user interface (RTUI) based on QualNet [1] has been designed and implemented to provide human users a way to dynamically revise the network parameters during the simulation process. Nonetheless, RTUI will import some delays into the simulation process caused by transmission and parsing, which will influence the accuracy of simulation results, especially for sensitive networks. Meanwhile, it is unprocurable for RTUI to modify several parameters in different re-configuration packets at the same time.

In this paper, a programmable dynamic simulation interface (PDSI) is further proposed. With the aid of PDSI, each of the pre-programmed re-configuration commands is marked with a time label, and then all the commands are encapsulated as packets and sent to the simulator in advance. The re-configuration command is pre-cached in the event queue based on the time label, and precisely implemented in terms of pre-programmed order, which eliminates the delay overhead of command transmission for the parameter changing of network model. Simulation results demonstrate the efficiency of the proposed PDSI compared to the RTUI.

The rest of the paper is organized as follows. Section II describes the programmable dynamic simulation interface in detail. In Section III, the proposed PDSI approach is evaluated by intensive experiments. Finally, the conclusions are drawn in Section IV.

Manuscript received October 31, 2016. This research was supported in part by National Science Foundation of China (No.61671062, No.61201180), Beijing Natural Science Foundation (N0.4132055), The Science and Technology Project of State Grid Corporation of China “Research on the key technologies for co-simulation of power grid and communication grid”, and Mid-career Researcher Program through NRF grant funded by the MEST (2016R1A2B4013388). Corresponding author: Peng Gong.

Haiqiao Wu is with the Beijing Institution of Technology, Beijing, China(whqjoe@gmail.com).

Lining Han is with the Beijing Institution of Technology, Beijing, China(lininghan@bit.edu.cn).

Li Huang is with the State Grid Global Energy Interconnection Research Institute, Nanjing, China(huangli@geiri.sgcc.com.cn)

Duk Kyung Kim is with the Inha University, Nam-gu Incheon City, Korea(kdk@inha.ac.kr).

Peng Gong is with the Beijing Institution of Technology, Beijing, China(phone:+86-010-6891-4850;fax:+86-010-6891-4850;e-mail:penggong@bit.edu.cn).

II. PROPOSED PROGRAMMABLE DYNAMIC SIMULATION INTERFACE

A. The Framework of PDSI

As shown in Fig. 1, the proposed PDSI framework is composed of GUI, QualNet-based simulator and PSDI. The GUI provides human users with a convenient configuration terminal to generate network re-configuration commands. The QualNet-based simulator is used to simulate the target networks. As the external interface of QualNet, the PSDI is an essential component in the framework, which combines the GUI with QualNet-based simulator together through UDP socket and is responsible for parsing configuration packets into corresponding recognizable messages by QualNet and pre-caching them into QualNet’s event queues.

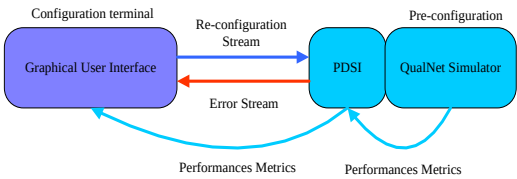


Figure 1. The framework of PDSI

Design of PDSI

Figure 2 shows the packet formats for PDSI, including re-configuration packet, metric packet, and error packet, which are denoted by the segment of packet type. The re-configuration packets are generated from the GUI to modify the parameters of network model, which consist of the packet type, the packet length, the node identification, the model type, the execution time, and the configuration parameters. Based on different network models, we can define different structures of the configuration parameters, which are distinguished by the model type. We can utilize some specific metric packets to acquire network performances, e.g., statistics of received or transmitted packets, packet error rate, and delay. A metric packet contains the packet type, the packet length, the node identification, the model type, and performances metrics, which is fed back to the GUI. PSDI also can indicate the error information through the error packets, when the re-configuration packets coming from the GUI are incorrect.

Re-configuration Packet Format					
Packet Type	Packet Length	Node Id	Model Type	Execution Time	Configuration Parameters

Metric Packet Format				
Packet Type	Packet Length	Node Id	Model Type	Performances Metrics

Error Packet Format				
Packet Type	Packet Length	Node Id	Error Type	Error Information

Figure 2. Packet formats of PDSI

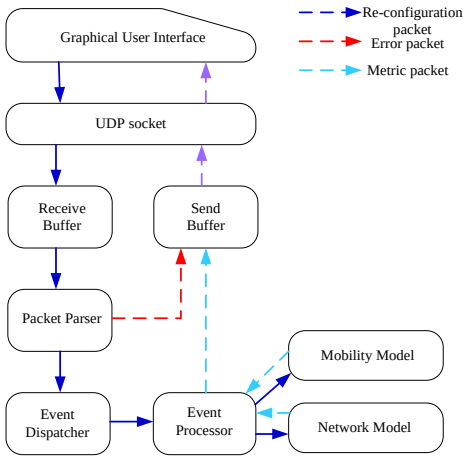


Figure 3. The working procedure of PDSI

The working procedure flowchart of PDSI is shown in Fig. 3.

- Via the GUI, users can pre-program a serial of re-configuration packets by hand or reading files. Once receiving the re-configuration packets, PSDI stores them in the receive buffer.
- When there are some re-configuration packets in the receive buffer, the packet parser is informed to interpret the re-configuration packets. If the re-configuration packets are incorrect, the packet parser returns the corresponding error packets to the send buffer. Otherwise, it translates the re-configuration packets into the messages and pre-caches them into the event queue based on the time label in all the re-configuration packets, which represents the execution time of the messages in the simulation. Fig. 4 shows the pre-caching process of PDSI based on the time label, which is different from the procedure of RTUI as shown in Fig. 5. In case of RTUI, all the messages are cached into the event queue at t_0 , which represents the current simulation time.
- As soon as the current simulation time equals to the value of time label in re-configuration packets, the corresponding events are assigned to the event processor by the event dispatcher.
- The event processor modifies the parameters of the network model or the mobility model based on the re-configuration parameters extracted from the messages.
- In order to acquire the network performances, different types of timer events are scheduled by the event processor periodically. And the processor extracts corresponding parameters from network models or mobility models based the type of the scheduled timer events. The extracted parameters are encapsulated as a metric packet to be put into the send buffer.
- The packets in the send buffer are fed back to the GUI via the UDP socket.

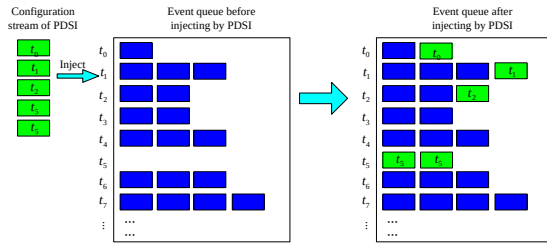


Figure 4. Pre-caching process of PDSI

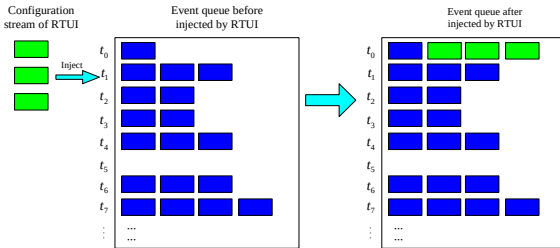


Figure 5. Caching process of RTUI

III. EXPERIMENTS AND RESULTS

A. Experiment setup

The experimental test-bed runs on QualNet version 5.1. And the parameters of the GUI client and the simulator are list in Table 1. The length of cable between the client and server is 3m. The proposed PDSI is compared to the RTUI in [8]. We test the single trip delay time between the GUI and the QualNet-based simulator, which is the delay introduced into simulation process by the RTUI and the delay of human action, e.g., time for typing the reconfiguration command, is neglected.

TABLE I
SIMULATION PARAMETERS

	GUI client	Simulator
Operation system	Windows7	Ubuntu 12.04 LTS
CPU	Intel(R) Core(TM) i7	Intel(R) core(TM) i5-3470
RAM	4GB	4GB

B. Experiment results

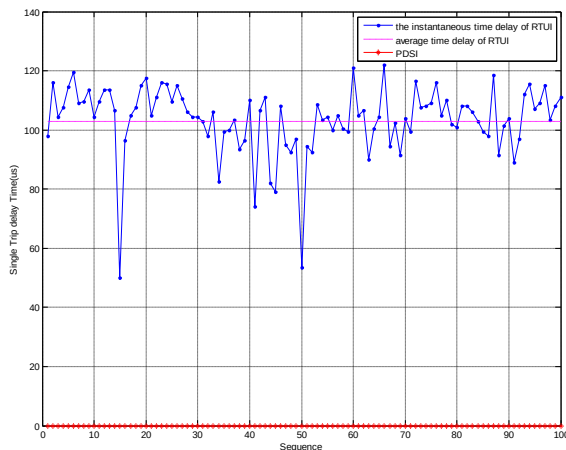


Figure 6. Single trip delay between GUI and QualNet

Fig. 6 shows the single trip delay time between the GUI and the QualNet-based simulator. By neglecting the human action delay time, the single trip delay time is composed of propagation time and processing time. The experiment result demonstrates that the average single trip delay time of RTUI is around 103us. However, as shown in Fig.6, owing to the pre-caching and time label, the transmission delay of re-configuration commands in PDSI is always zero. In addition, the human action delay time actually is much longer than the single trip delay time. Thus, if a large amount of re-configuration commands are needed to be executed, the fidelity of network simulator will be destroyed in the case of RTUI. Besides, the proposed PDSI provides a way to realize programmable simulation by pre-programming a set of re-configuration commands, which will be executed on the basis of the pre-programmed strategy. Therefore, the proposed PDSI is more efficient and practical for network model evaluation and testing.

IV. CONCLUSIONS

In this paper, a programmable dynamic simulation interface (PDSI) is proposed and implemented for the QualNet-based network evaluation and testing. Based on the proposed PDSI, the delay overhead, which is resulted from the re-configuration commands transmission, can be eliminated, since the commands with time labels are pre-cached in the event queue and implemented precisely based on the time labels. Simulation results has proved that the proposed PDSI is more efficient compared to the RTUI, especially for the case of a large amount of parameters changing for the large scale network models.

REFERENCES

- [1] P. Gong, M. Li, J. Kong, P. Li, D. K. Kim, "An interactive approach for QualNet-based network model evaluation and testing at real time," in *Proc. ICACT ' 16th*, 2014, paper 10.1109, p. 978-982.
- [2] M. Salman, M. A. Aziz, M. A. Shah, "Simulation based evaluation of DDH-MAC protocol for Cognitive Radio Networks," in *Proc. ICAC ' 22nd*, 2016, paper 10.1109, p. 516-521.
- [3] W. Liu, Xi Wang, W. Zhang, L. Yang, C. Peng, "Coordinative simulation with SUMO and NS3 for Vehicular Ad Hoc Networks," in *Proc. APCC ' 22nd*, 2016, paper 10.1109, p. 337-341.
- [4] A. Kheaksong, A. Prayote, W. Lee, "Performance evaluation of smart grid communications via network simulation version 3," in *Proc. ECTI-CON ' 13th*, 2016, paper 10.1109, p.1-5.
- [5] K. Y. Bendigeri, J. D. Mallapur, "Node placement in Wireless Sensor Network: An energy approach," in *Proc. RAECE 2015*, paper 2015.7510223, p. 40-45.
- [6] A. Alvarez, R. Orea, S. Cabrero, "Limitations of network emulation with single-machine and distributed ns-3," in *Proc. ICST ' 3rd*, 2010, paper 10.4108, p. 67.
- [7] G. O'Reilly, "Network capacity analysis using dynamic simulation," in *Proc. Telecommunications Network Strategy and Planning Symposium ' 13th*, 2008, paper 10.1109, p. 1-5.
- [8] T. Ye, H. T. Kaur, S. Kalyanaraman, M. Yuksel, "Large-Scale Network Parameter Configuration Using an On-Line Simulation Framework," *IEEE/ACM Transactions on Networking*, vol 16, pp. 777-790, Aug. 2008.



Haiqiao Wu received the BS and MS degrees in Mechatronical Engineering from Beijing Institute of Technology, Beijing, China, in 2014 and 2016 respectively. In September 2016, he attended in Beijing Institute of Technology for the Ph.D. degree. His research interests include network simulation, resource management in wireless systems, cognitive radio, 5G wireless networks and so on.



Lining Han received the BS and MS degrees from Electronic Engineering Institute of PLA. Now he is a PHD candidate in the School of Mechatronical Engineering, Beijing Institute of Technology. His research interests include network simulation and emulation, and the next generation wireless systems such as MIMO, Cognitive radio and so on.



Li Huang received the MS Degree in Software Engineering from University of science and technology of China, from 2011 to 2014, she worked in China Electric Power Research Institute and from 2014 worked in State Grid Global Energy Interconnection Research Institute. Her research interests include Automation of Electric Power Systems, Electric Power Information and Communication and so on.



Meng Song received the BS degree in Mechatronical Engineering from Beijing Institute of Technology in 2014, and now he is a MS candidate in School of Mechatronical Engineering, Beijing Institute of Technology. His research interests include wireless network simulation and emulation and network security modeling in mobile wireless networks.



Duk Kyung Kim received the B.S. degree in Electrical Engineering from Yonsei University, Seoul, Korea, in 1992, and the M.S. and Ph.D. degrees from the Korea Advanced Institute of Science and Technology (KAIST), in 1994 and 1999, respectively. From 1999 to 2000, he was a postdoctoral researcher at the Wireless Laboratories, NTT DoCoMo, Japan. From 2000 to 2002, he worked at R&D center, SK Telecom, Korea and involved in the standardization in 3GPP and also in 4G system development. In March 2002, he joined Inha University, Korea. His research interests include system performance evaluation at link/system level, radio resource management, and multi-media provision in wireless systems and next generation wireless systems such as 3GPP LTE, Wibro, UWB, MIMO, Cognitive radio and so on.



Peng Gong received the BS degree in Mechatronical Engineering from Beijing Institute of Technology, Beijing, China, in 2004, and the MS and Ph.D. degrees from the Inha University, Korea, in 2006 and 2010, respectively. In July 2010, he joined the School of Mechatronical Engineering, Beijing Institute of Technology, China. His research interests include link/system level performance evaluation and radio resource management in wireless systems, information security, and the next generation wireless systems such as 3GPP LTE, UWB, MIMO, Cognitive radio and so on.