

Design of Large Scale Network Simulator using Device Emulator for Internet of Things

Bumho Kim, Haeyong Kim, Seon-Tae Kim

ETRI(Electronics and Telecommunications Research Institute), Daejeon, Korea

mots@etri.re.kr, haekim@etri.re.kr, stkim10@etri.re.kr

Abstract— The term “Internet of Things” (IoT) denotes the interconnected of embedded devices designed to leverage data gathered by sensors. These devices are interconnected to transmit information and control instruction via the internet infrastructure. The development of an IoT system is a complex process due to the large scale and widely distribution of deployed wireless node. The simulator provides infrastructure to easily test and debug the algorithms of the IoT applications before they are to be deployed in actual environment. In this paper, we propose the emulation based network simulator architecture which focus on the large scale IoT system. With the proposed simulator architecture, the IoT application developers can reduce cost by cutting the amount of nodes needed for application test and shortens development time required for deploying a large scale IoT system.

Keywords— Network Simulator, Internet of Things, Device Emulator, Wireless Sensor Network, Smart Cities

I. INTRODUCTION

The term “Internet of Things” (IoT) denotes the interconnected of embedded devices designed to leverage data gathered by sensors. These devices are interconnected to transmit information and control instruction via the existing internet infrastructure [1]. The rapid development of IoT technology has triggered the extension of services in healthcare, public safety, connected mobility, connected home, and smart cities.

IoT applications require the deployment of a large number of smart interconnected devices to cooperatively transmit useful measurement information and control the environment. The development of an IoT system is a complex process due to the large scale and widely distribution of deployed wireless node. The simulator provides infrastructure to easily test and debug the algorithms of the IoT applications before they are to be deployed in actual environment [2, 3].

In this paper, we propose the emulation based network simulator architecture which focus on the large scale IoT system. By using device emulator, the proposed simulator evaluate and analyze the various components of the IoT applications. To enhance scalability of the pre-deployment simulation, the proposed scheme provides the network environments for testing the interaction occurring within the IoT application among nodes. With the proposed simulator architecture, the IoT application developers can reduce cost by cutting the amount of nodes needed for application test and

shortens development time required for deploying a large scale IoT system.

For testing the IoT application on simulator, we use the NanoQplus OS which is a light-weight operating system designed for IoT devices. The main features of NanoQplus are the support for multithreading and IETF(Internet Engineering Task Force) standard network stack such as IPv6, 6LoWPAN(IPv6 over Low-Power Wireless Personal Area Networks), and RPL(Routing Protocol for Low Power and Lossy Networks).

The rest of the paper is organized as follows. Section 2 addresses the IoT application development scenario. In Section 3, we present the NanoQplus operating system. In section 4, we propose our scheme and define the system model of the proposed method. Finally, we provide concluding remarks on our scheme in section 5.

II. IOT APPLICATION DEVELOPMENT SCENARIO

The IoT application may consist of thousands or even millions of smart devices interconnected via wireless network. The development of an IoT system is a complex and challenging process [4, 5].

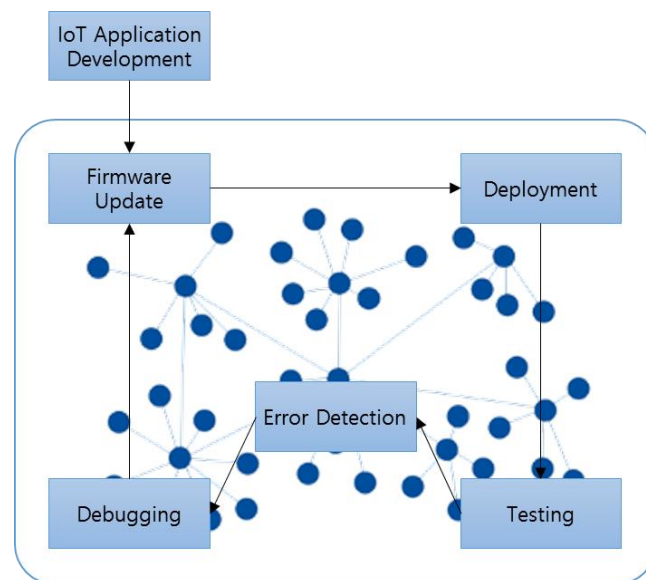


Figure 1. Use Case Scenario of IoT Application Development

Figure 1 shows a possible reference development workflow for large scale IoT applications. After developing the IoT application, the next thing is to update the software on the nodes and deploy them to target environment. If an error occurs in actual environment, it is necessary to collect the nodes, fix the bugs in software, update the software configuration on the nodes, and deploy them. Deployment of IoT infrastructure and setup of wireless networks are expensive and very costly in time.

By using simulator the development time of IoT application can be shortened. The software is uploaded on the virtual node and tested on environment before they are to be deployed.

III. NANOQPLUS OPERATING SYSTEM

The key feature of IoT system is that constrained devices with limited computation and battery capacity are interconnected via the existing internet infrastructure. It requires energy efficiency operating system with IPv6 network support [6, 7].

NanoQplus, which is developed by ETRI, is a lightweight operating system for IoT device. It supports a various IoT hardware and IPv6 network stack.

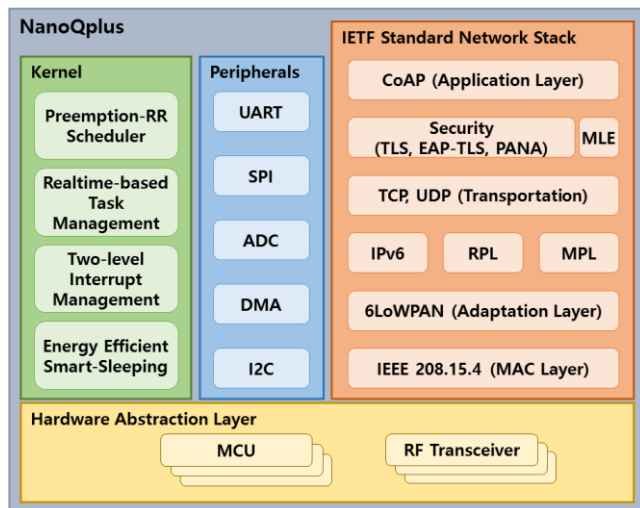


Figure 2. NanoQplus OS Structure

Figure 2 shows the overall structure of the NanoQplus operating system. NanoQplus helps developers to easily and reliably develop the complex application by supporting multi-threading. Moreover, it also supports IPv6 network stack and can develop IoT system that requires high-reliable connection and large-scale network. The main features of NanoQplus operating system are as follows.

A. Multithreading

Running the complex application as a single task in a sequential manner is nearly impossible due to the complexity of switching and processing. The NanoQplus supports user-level multithreaded programming on the complex application. It can divide the complex tasks into multiple threads and proceed while maintaining state of each divided thread separately. As a result, multithreaded application continues to

run even if part of it is blocked or performing a lengthy operation. Moreover, threads share the memory and the resources of the process to which they belong. It is more economical to create and context switch threads.

B. Smart Sleep

The NanoQplus kernel checks the current thread state and the list of scheduled tasks, and automatically switches to sleep mode, even if the users do not call the sleep mode API explicitly in the application. It provides means to decide whether or not to allow a packet to be received in a sleep mode.

C. IETF Standard Protocol

The NanoQplus operating system provides IEEE802.15.4 MAC based IPv6 internet communication by supporting the LoWPAN IETF standard. It also offer the TCP and UDP transport layer for end-to-end communication in the network. It is possible to communicate between IoT devices equipped with a web server application and a PC via web browser. Moreover, it supports the IETF RPL standard protocol enabling efficient routing paths for P2MP(Point-to-Multipoint) and MP2P(Multipoint-to-Point) traffic patterns in LLNs (Low Power and Lossy Networks). Supporting the IETF standard protocols enables enlarging the network in a multi-hop structure and guaranteeing the device reachability and reliability.

IV. LARGE SCALE NETWORK SIMULATOR

In this paper, we propose network simulator which focus on the large scale IoT system. Figure 3 shows the network simulator structure using device emulator.

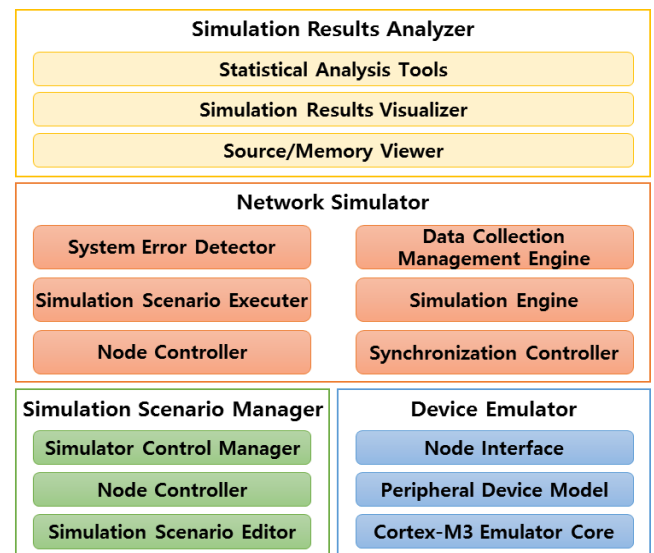


Figure 3. Network Simulator Structure

The device emulator module consists of a high speed emulator for low-power ARM Cortex-M3 MCU, an emulator for the peripheral devices, and the node interface for network simulator. It allows the application to be executed in a virtual environment on a PC.

The network simulator module evaluate the IoT applications using the device emulator in a large scale simulation environment. Moreover, it provides the function to control the simulation operation for time synchronization among the virtual nodes and detect the error in advance.

The simulation scenario manager provides the configuration functions of various environment changes and attributes such as a node mobility and state to improve the accuracy of network simulation.

In order to pre-verification of the IoT system the simulation results analyzer provides the visual display of network simulation result and the execution result of the emulator on virtual node.

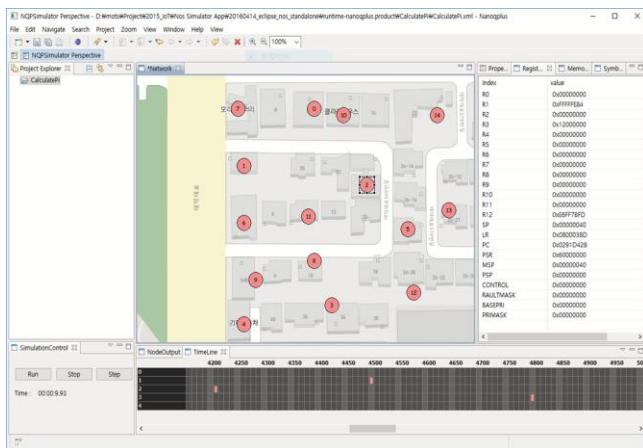


Figure 4. Network Simulator Screenshot

Figure 4 show a screenshot of network simulator with some emulation nodes started. The first step in testing the IoT application using a network simulator is to create and place the device node, set the desired simulation environment, create a simulation scenario, and test the user application. Next step is to analyze the results of the execution, check the errors, and correct the application until there are no more errors.

V. CONCLUSIONS

In this paper, we proposed the large scale network simulator architecture using a high speed emulator for low-power ARM Cortex-M3 MCU. Through the use of the device emulator, the proposed simulator evaluates and analyses the IoT application. In testing the large scale IoT application on network simulator, we use the NanoQplus OS which is a lightweight operating system designed for IoT devices.

To improve the scalability of the simulation, the proposed architecture provides the network environments for testing the interaction that occurs among the IoT nodes. With the proposed simulator architecture, the IoT application developers can reduce costs and time development time required to implement a large scale IoT system.

As future work, we plan to evaluate the performance of network stack including IEEE 802.15.4e standard and power consumption for IoT system.

ACKNOWLEDGEMENT

This work was supported by the ICT R&D program of MSIP/IITP. [1711028349, Development of integrated development solution supporting low power OS for lightweight embedded devices]

REFERENCES

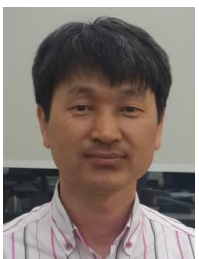
- [1] Y. Song, B. Han, X. Zhang, and D. Yang, "Modeling and simulation of smart home scenarios based on Internet of Things." 2012 3rd IEEE International Conference on Network Infrastructure and Digital Content. IEEE, 2012, p. 596-600.
- [2] G. Brambilla, M. Picone, S. Cirani, M. Amoretti, and F. Zanichelli, "A simulation platform for large-scale internet of things scenarios in urban environments," in Proc. of the First International Conference on IoT in Urban Space, 2014, p. 50-55.
- [3] S. N. Han, G. M. Lee, N. Crespi, K. Heo, N. V. Luong, M. Brut, and P. Gattellier, "DPWSim: A simulation toolkit for IoT applications using devices profile for web services." Internet of Things (WF-IoT), 2014 IEEE World Forum on. IEEE, 2014, p. 544-547.
- [4] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with cooja." Proceedings. 2006 31st IEEE Conference on Local Computer Networks. IEEE, 2006, p. 641-648.
- [5] B. Musznicki and P. Zwierzykowski, "Survey of simulators for wireless sensor networks." International Journal of Grid and Distributed Computing 5.3, 2012, p. 23-50.
- [6] E. Baccelli, O. Hahm, M. Günes, M. Wählisch, and T. C. Schmidt, "OS for the IoT-Goals, Challenges, and Solutions." Workshop Interdisciplinaire sur la Sécurité Globale (WISG2013). 2013.
- [7] E. Baccelli, O. Hahm, M. Günes, M. Wählisch, and T. C. Schmidt, "RIOT OS: Towards an OS for the Internet of Things." Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on. IEEE, 2013, pp. 79-80.



Bumho Kim received the BS degree in computer science from Sogang University in 2000 and MS degree in information technology from Information Communication University in 2002, respectively. Currently, he is a senior researcher in the Embedded SW Research Department at Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea. His research interests include embedded software, multimedia and distributed system.



Haeyong Kim received the B.S and M.S in Computer Science & Engineering from Seoul National University in 2004 and 2006, respectively. He is currently a researcher in Electronics and Telecommunications Research Institute (ETRI), Korea. He is one of developer of NanoQplus operating system. His research interests are lightweight operating system, embedded software, and sensor networks.



Seon-Tae Kim received the B.S, M.S and Ph.D. degree in the department of Electrical and Electronics Engineering from KAIST, Seoul National University and Korea University, Korea in 1997, 2000 and 2012, respectively. In February 2000, he joined the in the real-time multimedia team at the Electronics and Telecommunications Research Institute (ETRI), Korea. Since 2011, he has been a Director in the department of Embedded SW Research. His research interests are video compression, multimedia streaming, image processing, lightweight OS and sensor networks.