TELKOMNIKA, Vol.15, No.4, December 2017, pp. 1875~1882
ISSN: 1693-6930, accredited **A** by DIKTI, Decree No: 58/DIKTI/Kep/2013
DOI: 10.12928/TELKOMNIKA.v15i4.5725

Pervasive Device and Service Discovery Protocol in Interoperability XBee-IP Network

Sabriansyah Rizqika Akbar*, Helmi Nizar, Wijaya Kurniawan,
Mochammad Hannats Hanafi Ichsan, Issa Arwani
Faculty of Computer Science, University of Brawijaya, Jl. Veteran Malang, East Java, Phone: +62 341 551611/ Fax +62 341 565420
*Corresponding author, e-mail: sabrian@ub.ac.id[1], helminizar94@gmail.com[2], wjaykurnia@ub.ac.id[3], hanas.hanafi@ub.ac.id[4], issa.arwani@ub.ac.id[5]

Abstract

The Internet of Things (IoT) communication protocol built over IP and non-IP environment. Therefore, a gateway device will be needed to bridge the IP and non-IP network transparently since an IoT user is more likely to concern on the service provided by the IoT device, rather than the complexity of the network or device configuration. Since today ubiquitous computing needs to hide the architectural level from it users, the data & information centric approach was proposed. However, the data & information centric protocol is having several issues and one of them is device and service discovery protocol over IP & non-IP network. This paper proposed a pervasive device and service discovery protocol that able to work in interoperability of the IP and non-IP network. The system environment consists of a smart device with XBee Communication as the non-IP network that will send the device and service description data to the IP network using WebSocket. The gateway will able to recognize the smart device and sent the data to the web-based user application. The user application displayed the discovered devices along the services and able to send the control data to each of the smart devices. Our proposed protocol also enriched with the smart device inoperability detection by using keep-alive tracking from the gateway to each of the smart devices. The result showed that the delay for the user application to detect the smart device in the XBee network is around 10.13 ms delay, and the service average delay requested by the user application to each of the devices is 2.13 ms.

Keywords: *pervasive, interoperability, websocket, Xbee*

1. Introduction

Interoperability and interconnecting devices that able to adapt autonomously is a central issue in the Internet of Things (IoT) [1]. In the pervasive and ubiquitous environment, providing on-the-fly communication services is recently considered as an emerging field of technology and research [2]. With the heterogeneous pervasive environment, the communication protocol will be varied and make it more complex to its users [3]. The pervasive and ubiquitous computing goal is to hide this complexity to it users by creating a transparent network and make the user more focus on the application [4]. At the application level, a system that provides context awareness become more important since the user only cares about what device and service provided in the pervasive environment beside of the technology that relies on it. There are several pervasive application-level protocols such as UPnP [5], Alljoyn [6], CoAP [7], etc.

The pervasive device and service discovery protocol is also implemented in wireless Machine-to-Machine (M2M) communication. There are several wireless standards in M2M, and 802.15.4 becomes more popular for M2M communication since it has been standardized by IEEE and enhanced the low cost and low power consumption features. XBee [8] is an upper layer communication device based on 802.15.4 technology and now widely used in wireless network field [9]-[11]. XBee is a non-IP communication device that has the ability to self-organized their network but needs to be integrated with a gateway to make it able to communicate with the TCP/IP network.

European Telecommunication Standard Institute (ETSI) has been developed a standard topology for M2M contained Network (TCP/IP) and Device and Gateway Domain (Non-IP Communication) [12]. The border between the network and the device domain is separated by the gateway and have a different physical layer communication. In the TCP/IP communication,

Received August 15, 2017; Revised October 28, 2017; Accepted November 24, 2017

WebSocket becomes more popular since it enhanced the HTTP performance by adding real-time bi-directional, and persistent connection with the server [13]. WebSocket also popular to be used as control interface in smart technology, since it deployed above the widely used HTTP protocol, WebSocket able to be created with intuitive Graphical Utility Interfaces in the web browser [14], [15]. Our research proposed a gateway design and implementation for XBee and the TCP/IP network that able to connect seamlessly by providing the pervasive device and service discovery protocol. By using our method and architecture, the user will be able to more focus on the service provided by the XBee based devices and not bothered by the complexity of the network architecture and configuration.

2. Proposed Method

Our research using XBee [8] as a ZigBee device and will communicate with the WebSocket technology to perform the pervasive device and service discovery protocol. With topology shown in Figure 1. There is XBee area Network and the TCP/IP Network that will be communicated each other using a gateway. We provide 4 XBee act as smart device and configured as XBee Router device, gateway plugged with XBee act as coordinator (using Raspberry Pi), WebSocket server and web-based user application installed on a computer.

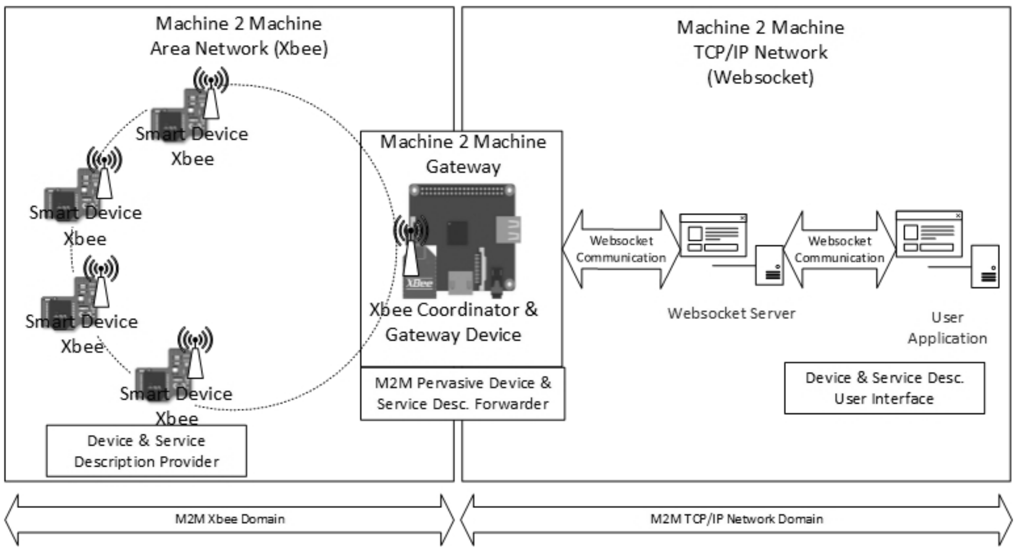


Figure 1. XBee & TCP/IP M2M Network

Our research proposed pervasive device and service discovery shown in Figure 2. At the first stage, the web-based user application will be connected with the WebSocket server to make the web user application always ready to receive data from the WebSocket server. If the gateway device turned on, it will be automatically connected with the WebSocket server. At the next stage, when there is a smart device around the gateway network with PAN ID 1AAA, smart devices will inform its presence to the gateway by sending broadcast information along with it device and service description presented in Figure 3. If the first smart device active, it will inform the gateway that the smart device is lamp the type is an actuator and provide turn on and turn off services. If the second device active, it will inform the gateway that the smart device is door as an actuator and have open, close, and lock services.

After the gateway received the broadcast information, the gateway will send a unicast acknowledgment packet to make the smart device change it state to listen to a service request. When the gateway received the device and service description, the gateway will forward the information to the WebSocket server and WebSocket server will push the device and service description to the web user application. The web user application will be display the graphical utility interface that matched with the device and service description sent by the smart device.

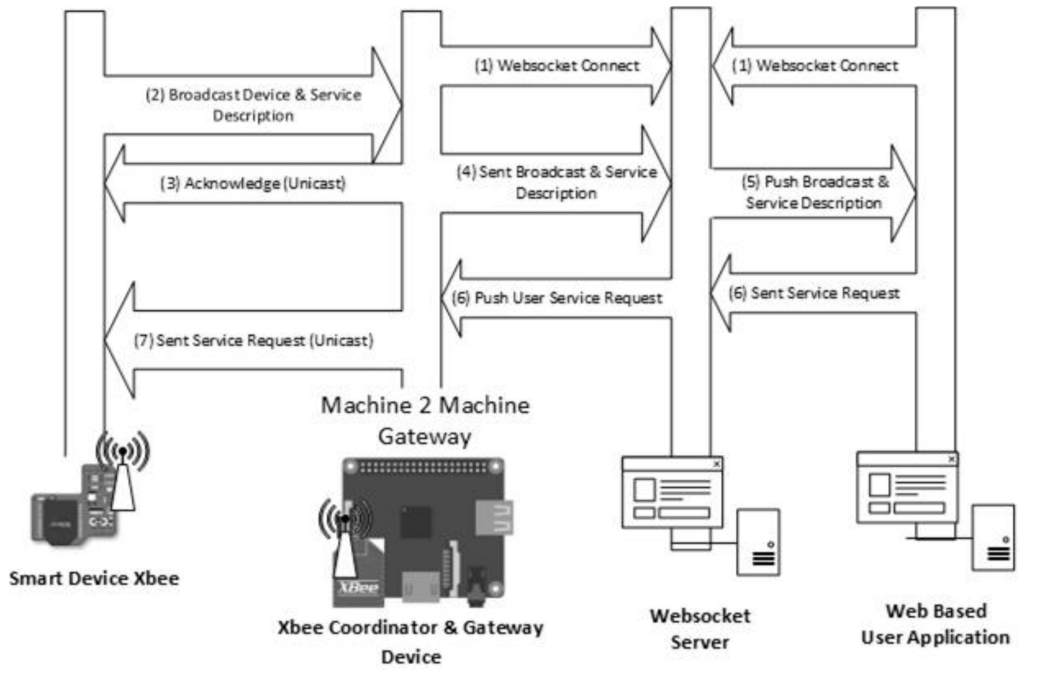

(1) Websocket Connect
(1) Websocket Connect
(2) Broadcast Device & Service Description
(3) Acknowledge (Unicast)
(4) Sent Broadcast & Service Description
(5) Push Broadcast & Service Description
(6) Push User Service Request
(6) Sent Service Request
(7) Sent Service Request (Unicast)
Machine 2 Machine Gateway
Smart Device Xbee
Xbee Coordinator & Gateway Device
Websocket Server
Web Based User Application

Figure 2. Proposed Pervasive Device & Service Discovery

```
First Smart Device: {"device":"Lamp","type":"actuator","service":["Turn On","Turn Off"]}
Second Smart device: {"device":"Door","type":"actuator","service":["Open","Close","Lock"]}
```

Figure 3. Device and Servi ce Description Data

When the interface displayed, user will be able to choose which device will be controlled and which service will be requested. Figure 4 and Figure 5 show the system entities flowchart contained of smart device, gateway, and the user application. Figure 4(a) explained the flowchart from smart device as the provider of the device and service discovery data. The smart device will broadcast the information to the gateway/coordinator and will call a receive handler to listen to the ACK message that will be sent from the gateway, the smart device will change it back to the broadcast device and service discover state if the ACK message not received in 30s.

The receive handler function is also used to receive service data requested by the user application. Figure 4(b) displayed the flowchart and show the process when the gateway forwards the pervasive device and service discovery data from the XBee network to the user application. Our research proposed the gateway method as a development from previous research [16] by added WebSocket protocol as the TCP/IP communication between the gateway and user application.

The gateway will run three threads which are XBee receive the callback, keep-alive messages, and WebSocket receive the callback. XBee receive the callback is used to listening package from the smart device in the XBee network. Gateway will receive device and service description data from the smart device and will be sent it to user application via WebSocket protocol. The keep-alive thread is use to detect an inactive smart device and inform user application that the smart device is unavailable. The keep-alive thread will send a “ping” message to the smart device periodically and if the smart device is not replied the message the smart device will be considered as an inactive node.

The WebSocket receive thread will handle the WebSocket receive communication to the WebSocket server and user application. The WebSocket receive thread is equipped with the duplication check feature to avoid the same smart device duplication. The WebSocket receive thread is able to recognize the smart device uniquely by the long address and the short address, If the smart device long and short address is already recognized by the gateway, it will not be published to the user application and the device and service discovery data will be ignored.

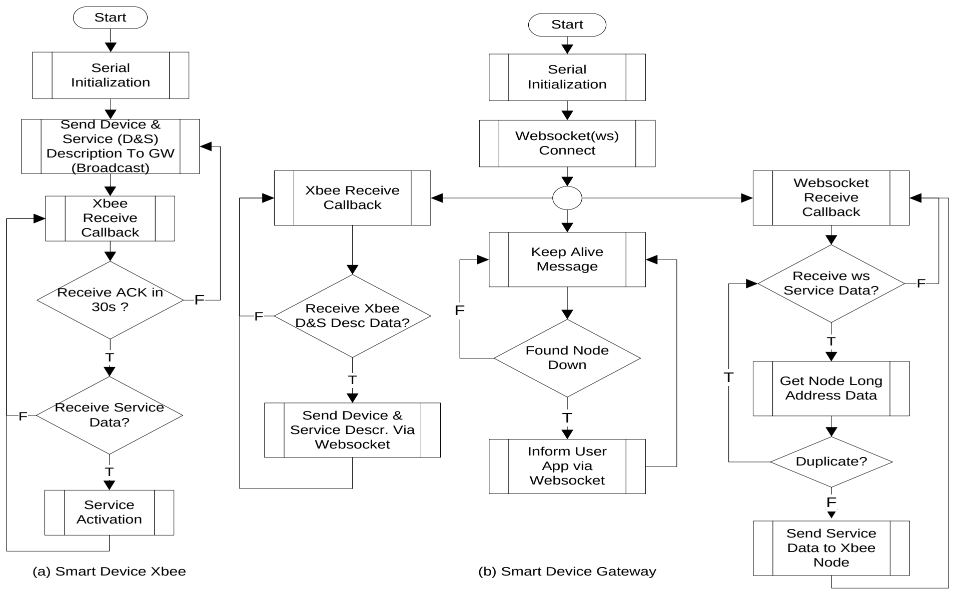


Figure 4. Smart Device and Gateway Flowchart

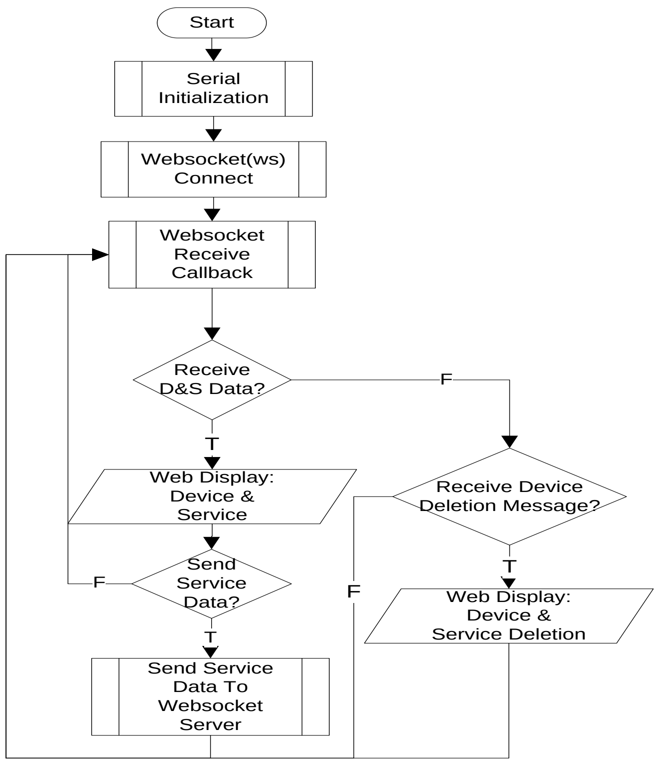
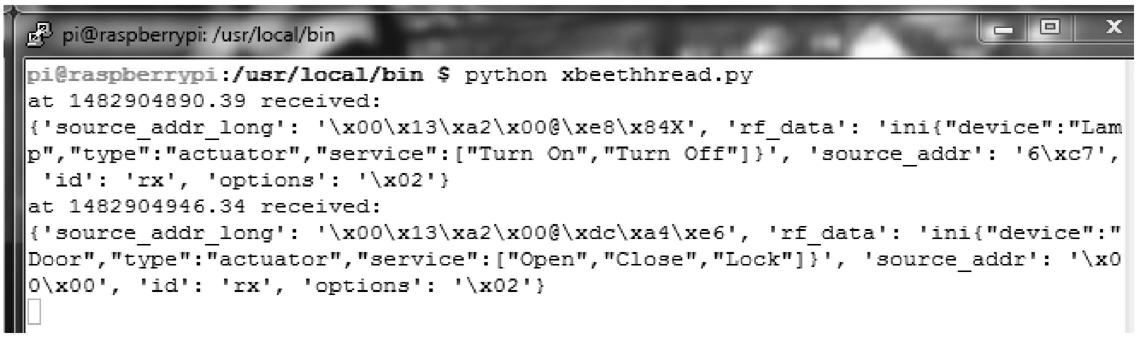

Start
Serial Initialization
Websocket(ws) Connect
Websocket Receive Callback
Receive D&S Data?
F
T
Web Display: Device & Service
Receive Device Deletion Message?
Send Service Data?
F
T
F
T
Web Display: Device & Service Deletion
Send Service Data To Websocket Server

Figure 5. User Apllication Flowchart

Figure 5 presenting the flowchart for user application. The user application interface is created in web-based environment and WebSocket protocol for communication with the gateway. The user application will receive device and service description data from the gateway and displayed on the web. The user will also have a device removal notification if there is unavailable smart device detected by the keep-alive process in the gateway. After the device and service description has been received by the user application, the user web browser will display each of the recognized devices and the user will be able to send a service request to the smart device via the gateway.

3. Result and Analysis

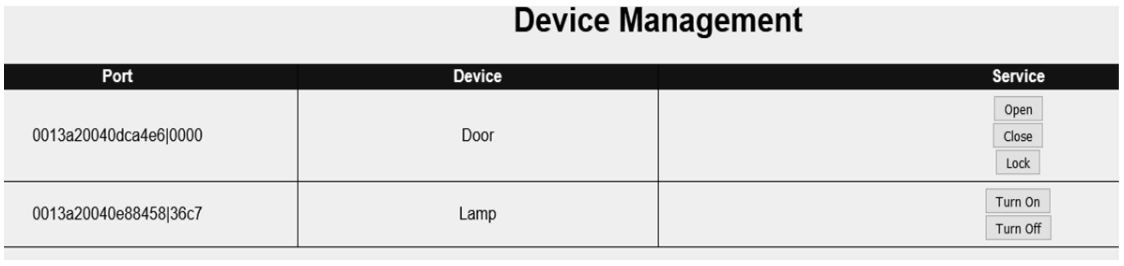
Smart device will broadcast its presence to the gateway and will be forwarded to the user application. Figure 6 displayed the gateway status when received the broadcast device and service description. The displayed information showed that there is two devices lamp and door that identified by each XBee source long address. After the gateway received the broadcast message, the gateway will send the information to the WebSocket server and push it to the user application interface.



```
pi@raspberrypi:/usr/local/bin $ python xbeethhread.py
at 1482904890.39 received:
{'source_addr_long': '\x00\x13\xa2\x00@\xe8\x84X', 'rf_data': 'ini{"device":"Lam
p","type":"actuator","service":["Turn On","Turn Off"]}', 'source_addr': '6\xc7',
 'id': 'rx', 'options': '\x02'}
at 1482904946.34 received:
{'source_addr_long': '\x00\x13\xa2\x00@\xdc\xa4\xe6', 'rf_data': 'ini{"device":"
Door","type":"actuator","service":["Open","Close","Lock"]}', 'source_addr': '\x0
0\x00', 'id': 'rx', 'options': '\x02'}
```

Figure 6. Gateway Broadcast Received

User application interface presented in Figure 7 and displayed two smart device which is lamp and door. The smart devices identified uniquely in the port field by the smart devices long address. The door device is identified in the 0013A200400DCA4E6 and the lamp device is identified in 0013A20040E88458. The user will be able to send the service data request for each device displayed in the service field. The door device has an open, close and lock services, while the lamp device provides turn on and turn off services.

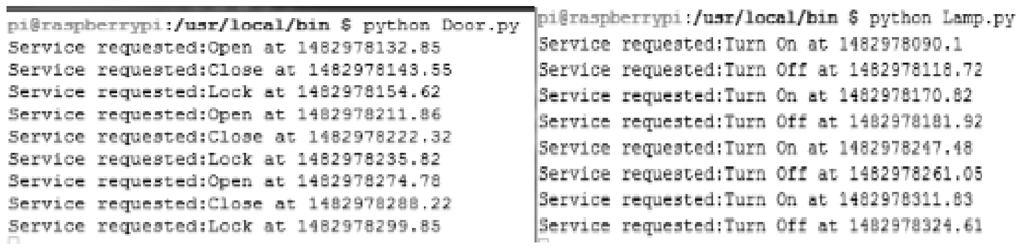


Device Management

Port	Device	Service
0013a20040dca4e6|0000	Door	Open Close Lock
0013a20040e88458|36c7	Lamp	Turn On Turn Off

Figure 7. User Application Interface

The smart device received the service requested by the user via gateway as shown in Figure 8. We provide time delay measurement to check the response time when the user sent the service request until the smart device receive the service requested.



```
pi@raspberrypi:/usr/local/bin $ python Door.py
Service requested:Open at 1482978132.85
Service requested:Close at 1482978143.55
Service requested:Lock at 1482978154.62
Service requested:Open at 1482978211.86
Service requested:Close at 1482978222.32
Service requested:Lock at 1482978235.82
Service requested:Open at 1482978274.78
Service requested:Close at 1482978288.22
Service requested:Lock at 1482978299.85
```

```
pi@raspberrypi:/usr/local/bin $ python Lamp.py
Service requested:Turn On at 1482978090.1
Service requested:Turn Off at 1482978118.72
Service requested:Turn On at 1482978170.82
Service requested:Turn Off at 1482978181.92
Service requested:Turn On at 1482978247.48
Service requested:Turn Off at 1482978261.05
Service requested:Turn On at 1482978311.83
Service requested:Turn Off at 1482978324.61
```

Figure 8. Service Received and Executed by the Smart Device

Our research conducted a two experiment scenario. The first scenario is to measure the delay when the smart device start to send the broadcast device and service information until it has been received by the user application. The second scenario is conducted by measuring the delay when the user application sent the service data to the smart device.

Table 1. Broadcast Time and Time Received Delay

No.	Delay (milisecond)					
	Smart Device-WebSocketServer		WebSocket server-User Application		Smart Device-User Application	
	Lamp	Door	Lamp	Door	Lamp	Door
1.	0.20	0.21	9.92	9.93	10.12	10.13
2.	0.20	0.22	9.92	9.93	10.12	10.15
3.	0.21	0.22	9.92	9.92	10.13	10.14
4.	0.21	0.22	9.92	9.92	10.13	10.14
5.	0.21	0.21	9.92	9.92	10.13	10.13
6.	0.20	0.21	9.98	9.92	10.18	10.13
7.	0.21	0.21	9.92	9.92	10.13	10.13
8.	0.21	0.21	9.92	9.93	10.13	10.14
9.	0.20	0.22	9.92	9.93	10.12	10.14
10.	0.20	0.22	9.93	9.92	10.13	10.14
Average	0.205	0.215	9.927	9.924	10.132	10.137

Table 1 displayed the delay information when the smart device broadcast its device and service description to the WebSocket server and user application. The result showed that the average delay between smart devices with the WebSocket server is 0.205 for the Lamp device and 0.215 for the door device. The average delay between the WebSocket server and user application is 9.927 ms for the lamp device and 9.93 ms for the door device. Figure 9 presented the delay measurement from the smart device to user application with average delay 10.137 ms. The second experiment scenario conducted is to measure the service delay between the user applications to the smart device. User application will be able to send the service data (such as turn on or turn off lamp) after the user application able to recognize each of the devices and service descriptions.

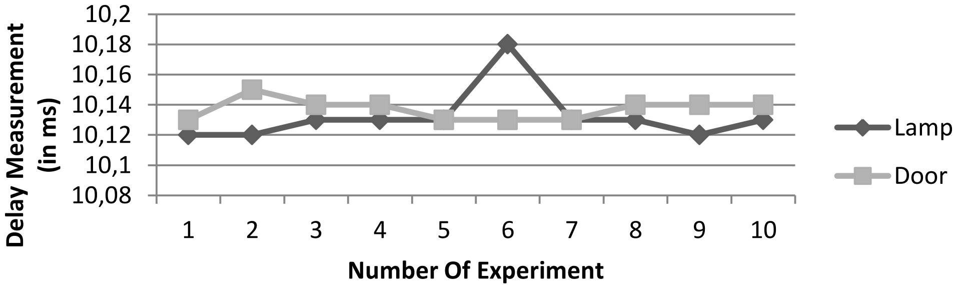


Figure 9. Smart Device-User Application Delay Measurement

Table 2. Service Delay

No.	Delay in Milisecond Delay Web Browser to Gateway (milisecond)	Delay Gateway to Device (milisecond)	Delay Web Browser to Device (milisecond)
1	2.268000126	0.059999943	2.328000069
2	2.269000053	0.059999943	2.328999996
3	2.281000137	0.059999943	2.34100008
4	2.271000147	0.059999943	2.33100009
5	2.282000065	0.059999943	2.342000008
6	2.27699995	0.059999943	2.336999893
7	2.282000065	0.059999943	2.342000008
8	2.282000065	0.059999943	2.342000008
9	2.279999971	0.060000181	2.340000153
10	2.282999992	0.059999943	2.342999935
11	2.282999992	0.060000181	2.343000174
12	2.290999889	0.059999943	2.350999832
13	2.286000013	0.059999943	2.345999956
14	2.375999928	0.070000172	2.446000099
15	2.286000013	0.059999943	2.345999956
16	2.29399991	0.060000181	2.354000092
17	2.289999962	0.059999943	2.349999905
18	2.290999889	0.060000181	2.351000071
19	2.292000055	0.059999943	2.351999998
20	2.296000004	0.059999943	2.355999947
Average:	2.288000011	0.060500002	2.348500013

Table 2 showed the experiment result for the service delay sent by the user application. The average delay from the web browser user application to the gateway is 2.288 ms. And the average delay from the gateway to the smart device is 0.060 ms. The average delay from the user application to the smart device is 2.34 ms and presented in Figure 10.

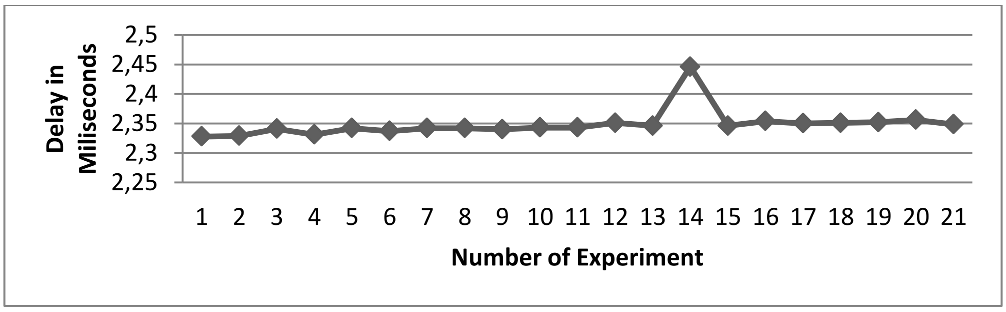


Figure 10. Service Delay between the User Applications to Smart Device

4. Conclusion

Our research successfully developed the pervasive and service discovery protocol in Interoperability XBee-IP network. Our system proposed XBee as smart device and the gateway that able to made interoperability into the WebSocket. The smart devices will be broadcast the pervasive and service description data into the gateway and the gateway will push the data to the user application with WebSocket protocols. The user application will be able to send the service data to the smart device via the gateway. Our research developed smart door device prototype with close, open, lock services and smart door lamp with turn on and turn off services. The user application will be displayed the device and each of the device services in the web browser and able to send the door and lamp services data to the smart device.

Our system enriched with the keep-alive features to detect if there is smart device that not able to function properly and inform the web user application if the smart device will be inaccessible. From the conducted experiment, the average delay when the smart device is recognized by the user application is around 10.13 ms delay. The average service delay when the user application sent service data to the smart device is around 2.34 ms delay. The pervasive device and service discovery protocol in the XBee-IP network interoperability able to made the transparent network and made the user more focusses on the services provided by the system.

Reference

[1] L. Atzori, A. Iera and G. Morabito. The Internet of Things: A survey. *Computer Networks.* 2010; IV(54): 2787-2805.

[2] F. Palmieri. Scalable service discovery in ubiquitous and pervasive computing architectures: A percolation-driven approach. *Future Generation Computer Systems.* 2012; 29: 693-703.

[3] Li Han, Huan-yan Qian. The Implementation of One Opportunistic Routing in Wireless Networks. *TELKOMNIKA.* 2015; 13(2): 460-468.

[4] S. Najar, M. K. Pinheiro and C. Souveyet,. *A new approach for service discovery and prediction on Pervasive Information System.* in International Conference on Ambient Systems, Networks and Technologies (ANT-2014),the 4th International Conference on Sustainable Energy Information Technology (SEIT-2014), Paris, 2014

[5] Open Connectivity Foundation, About UPnP, [Online]. Available: http://openconnectivity.org/upnp. [Accessed 2 May 2016].

[6] Allseen Alliance, AllJoyn Documentation, [Online]. Available: https://allseenalliance.org/framework/documentation. [Accessed 2 May 2016].

[7] C. Bormann, A. P. Castellani and Z. Shelby. CoAP: An Application Protocol for Billions of Tiny Internet Nodes*. IEEE*, 2012.

[8] Digi, Xbee, Digi, [Online]. Available: http://www.digi.com/lp/xbee. [Accessed 10 August 2016].

[9] A. H. Kioumars and D. L. Tang. *ATmega and XBee-Based Wireless Sensing*, in Proceedings of the 5th International Conference on Automation, Robotics and Applications, Wellington, New Zealand, 2011.

[10] N. Zulkifli, F. Harun and N. Azahar. *XBee Wireless Sensor Networks for Heart Rate Monitoring in Sport Training*. in 2012 International Conference on Biomedical Engineering (ICoBE) , Penang, Malaysia, 2012.

[11] R. L. Pascual, D. M. R. Sanchez, D. L. E. Naces and W. A. Nuñez,. *A Wireless Sensor Network Using XBee for Precision Agriculture of Sweet Potatoes (Ipomoea batatas).* in 8th IEEE International Conference Humanoid, Nanotechnology, Information Technology Communication and Control, Environment and Management (HNICEM), Cebu, Phillliphines, 2015.

[12] J. Holler, V. Tsiatsis, C. Mulligan, S. Karnouskos, S. Avesand and D. Boyle, *From Machine-to-Machine to the Internet of Things Introduction to a New Age of Intelligence, Massachusetts*: Elsevier, 2014.

[13] Lihong Geng, Liang Pan, Yiqiang Sheng, Zhichuan Guo. Towards Smooth and High-Quality Bitrate Adaptation for HTTP Adaptive Streaming. *TELKOMNIKA*. 2016; 14 (3): 904-915.

[14] T. Abdulrahman, O. Isiwekpeni, N. Surajudeen-Bakinde and A. Otuoze. Design, Specification and Implementation of a Distributed Home Automation System. *Procedia Computer Science*; 2016: 94: 473-478.

[15] L. Srinivasan, J. Scharnagl and K. Schilling. *Analysis of WebSockets as the New Age Protocol for Remote Robot Tele-operation*. in The International Federation of Automatic Control, Seoul, Korea, 2013.

[16] S. R. Akbar, W. Kurniawan, M. H. H. Ichsan, I. Arwani and M. T. Handono. *Pervasive Device and Service Discovery Protocol In XBee Sensor Network*. in 2016 International Conference on Advanced Computer Science and Information Systems, Malang, East Java, Indonesia, 2016.