

RESTful Web Service and Push Service based Management of BLE Beacon Network

YoungWook Cha*, DongMin Kim* and ChoonHee Kim**

*Department of Computer Engineering, Andong National University, Korea

**Department of Electronic & Information Communication Engineering, Daegu Cyber University, Korea
ywcha@andong.ac.kr, chkim@dcu.ac.kr

Abstract

This study suggests RESTful web service and push service based management of BLE beacon network that consists of beacons, gateways and manager. For management information exchange, RESTful web service based RBNM (RESTful web service for BLE beacon Network Management)-API was defined, and using RESTClient, the validity of the defined RBNM-API was verified. By implementing the BLE beacon network management system as a laboratory model, manager processing time was measured and analyzed according to database access methods and JSON parser types to process beacon configuration information. Websocket, web's push service technology was compared with existing polling and Comet technologies in terms of waiting time of event notification. Event notifications toward smart devices tested and notification delays were measured with GCM3.0-API and RBNM-API.

Keywords: ble beacon; management; web service; event notification; push service

1. Introduction

From new venture companies to large corporations, companies are introducing services using BLE (Bluetooth Low Energy) beacons, engaging in fierce competition. Beacons show special strength in online and off-line (O2O) service marketing but as the number of managed beacons increases, problems such as protocol interference between similar devices and power consumption are rising [1]. This study suggests management function and management configuration of BLE beacon network that consists of gateways and manager. For management information exchange, RESTful web service [2] based RBNM (RESTful web service for BLE beacon Network Management)-API was defined, and using RESTClient [3], the validity of the defined RBNM-API was verified.

By implementing the BLE beacon network management system as a laboratory model, manager processing time was measured and analyzed according to types of JSON parser to process RBNM-API and methods of database access to process beacon configuration information transmitted from gateway. We compared Websocket [4], web's push service technology with existing polling and Comet technologies in terms of waiting time of event notification. We also tested event notifications toward smart devices and measured notification delays with GCM3.0-API [5] and RBNM-API.

2. Related Research

2.1. Web Service based Management

SNMP is the Internet management protocol using the connectionless transport layer. Thus, it can hardly respond effectively to an increase in network size and management information due to the limited data expression, limited message length and reliability. The network management research group of IRTF recognized web service based management as a solution for the limitation of SNMP (Simple Network Management Protocol), transaction and security related issues. Twente University proved that web service was as good as SNMP in terms of bandwidth, memory use, CPU usage time and round-trip delay [6].

SOAP based web service was initiated for the inter-working of applications in a business environment. In contrast, RESTful web service was initiated for the purpose of allowing Internet service providers to offer data conveniently to application developers. SOAP based web service is developed through the strict standard and well-equipped infrastructure of W3C. RESTful web service, which requires only the basic Internet standards, can access those resources allowing for various expressions (XML, JSON, HTML, image, *etc.*) only with the basic types of HTTP. Furthermore, it's another distinctive feature is that the status information of clients is not managed at the server. With this simplicity and convenience of development, such Internet giants as Google, Amazon, Yahoo, Twitter, *etc.* are opening their own information resources through RESTful web service based API and 85 percent of open-type API at Amazon is RESTful web service [2]. CoRE (Constrained RESTful Environments) working group of IETF proposed RESTful web transmission protocol that redesigned several functions of HTTP for M2M application in a small-sized device such as a sensor node [7].

2.2. Push Service

Polling technique is to deliver a message if requested by a client, while push technique is to deliver a message without request. Push system is comprised of a provider server, a push server, and smart devices. Provider server produces a message to notify and deliver it to the push server, while push server delivers a received message to the smart device. As for examples of push servers, there are GCM (Google Cloud Messaging) from Google, APNS from Apple, AOM (Always on Management) from SKT, and nPush supported by NHN [8]. GCM server returns register-ID to the smart device that requests registration of push service. Provider server delivers register-ID of smart device and message to push to GCM server in the use of JSON (JavaScript Object Notation)-based GCM-API. GCM server pushes received message to the smart device that is relevant to register-ID [5]. AOM platform has been developed to solve excessive traffic from an increased amount of application with push service and fast-draining battery. NHN's nPush provides a comprehensive interface on APNS, C2DM, and AOM platforms. Li [9] compares push and polling as well as push techniques of each company, length of message, message format, and preservation period of message.

In the previous HTTP requesting/responding model, polling is that web browser requests data to server in a constant cycle. Comet is that server responds when data is created upon request from a web browser. Polling and Comet techniques suffer network overhead making it difficult to support real-time service according to stand-by time of data produced in the server [10]. In order to solve such issues, W3C has defined Websocket API of HTML5, a bi-directional communication technology between web-server and web browser, and IETF has standardized protocol with RFC 6455 [4].

2.3. BLE Technology

If Bluetooth 3.0 puts its purpose on increasing transmission speeds, Bluetooth 4.0 (BLE) concentrated on power consumption reduction. BLE beacon is an indoor positioning technology that is based on radio signal and can communicate with devices up to 70m apart, has accuracy to distinguish units of 5 to 10cm, and has low power consumption. Because of these merits, BLE beacon is appropriate for implementation of IoT (Internet of Things) in which all devices are always connected [11]. The advantage of low power consumption in BLE beacons is appropriate for home automation, wireless medical devices, exercise sensors, and mobile payments. It is forecasted that the global market utilizing BLE beacon will expand from \$31.7 million in 2013 to \$216.7 million in 2018 [12]. Son [13] determined the operation status of large-scale beacons using beacon identifier values according to user location history in Android application. Study [14] developed and introduced beacon management dashboard, and managed location information of beacons by expressing them on a map using only unique IDs transmitted through BLE. Han [15] utilized BLE technology to design a system that sends push notifications for product information, and Lee [16] utilized a wearable device form's Bluetooth beacon to implement an indoor location-based monitoring system for dementia patients.

3. Function and Web Service for BLE Beacon Network Management

3.1. Configuration and Function for BLE Beacon Network Management

Installations of BLE beacons are increasing due to the advantages of low power consumption and highly accurate indoor positioning technology, and continuous maintenance and management of the installed beacons are required [1, 14]. Because BLE protocol transmits data to a limited distance, there is a problem of managing beacons in that administrator must communicate with the beacons directly by going to the places where the beacons are installed. BLE-GW that has the role of proxy agent and interworking function between Ethernet and BLE protocol is introduced to solve this problem. A proxy agent has the role of providing beacon information to BLE beacon manager instead of agent function of BLE beacon or setting the information required by the manager to the beacon. The management configuration of BLE Beacon network that is composed of manager, gateways, and agents is shown in Figure 1.

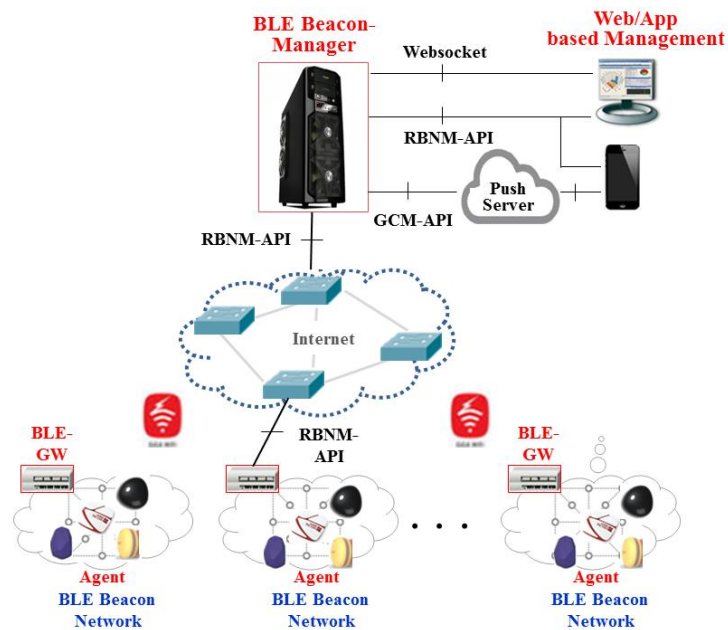


Figure 1. Configuration of BLE Beacon Network Management

Using RESTful web service (RBNM-API), BLE beacon network management information is exchanged between BLE beacon manager and BLE-GW, and BLE protocol is used to exchange management information between BLE-GW and beacons. BLE beacon transmits its information (UUID, Major, Minor, battery information, etc.) to BLE-GW each regular cycle (5 seconds) and BLE-GW gathers information from beacons within its domain and transmits gathered information to BLE beacon manager on a regular cycle (15 seconds). The manager uses RBNM-API which is commonly applied to web browser and hybrid app of management terminal to perform management of BLE beacon network and uses Google GCM-API for real-time event notification to smart devices. The main functions of BLE beacon manager that comprehensively performs configuration and fault management of BLE beacon network are as follows.

- Configuration management of beacon network
 - o Registration, setup, modification and query of beacon configuration information: beacon ID management (UUID, Major, Minor)
 - o Automation of beacon configuration management with beacon ID
 - o BLE beacon's battery residue check
 - o Registration, setup, modification and query of BLE-GW's configuration information
- Fault and event management of beacon network
 - o Real-time beacon fault detection (declared as faulty if signal is not detected 3 times in a row between beacon and BLE-GW)
 - o The primary event is notified when the battery level is under 5% and the secondary event is notified when it is fewer than 2%.
 - o Event notification in case of unauthorized changes on beacon information (UUID, strength, cycle, etc.)
 - o BLE-GW is declared as faulty when notification is not received 3 times in a row from BLE-GW.
 - o Real-time event notification to web browsers using Websocket or smart devices using GCM push service
 - o Generation and query of event or push log
- Management of administrators and push users
 - o Registration, modification or query of administrators
 - o Registration, setting, modification, and query of push users for smart devices where the event is notified

3.2. RESTful Web Service for BLE Beacon Network Management

RBNM (RESTful web service for BLE beacon Network Management)-API is defined for management information exchange between BLE beacon manager and management terminal or between manager and BLE-GW. Information form of RBNM-API is divided into group and table. Group represents collection of management objects with high association with simultaneous management and table is a list of entries composed of management objects [17]. RBNM-API uses GET, POST, PUT, and DELETE request types of HTTP message. On the URL of the request message, “manager or BLE-GW address/information type (table, group)/information name/” are commonly included. The representative request types and URLs of RBNM-API for management of BLE beacon network are shown in Table 1.

Table 1. Request Types, URLs and Status Codes of RBNM-API

Request type	URL (http://beacon-manager or ble-gw address)	Status code	
		ok	err
POST	/group/login or logout	200	400
POST PUT	/table/blegw/entry: addition or modification of BLE-GW	201 200	400
DELETE GET	/table/blegw/entry/indexes/blegwname/{blegwname}: deletion or detail query for a specific BLE-GW	204 200	400
POST PUT	/table/beacon/entry: addition or modification of a BLE beacon	201 200	400
DELETE GET	/table/beacon/entry/indexes/macaddress/{macaddress}: deletion or detail query for a specific beacon	204 200	400
GET	/table/eventlog/entry/indexes/starttime&endtime/{starttime&endtime}: event query for a specific time period	200	400
GET	/table/eventlog/entry/indexes/blegwname/{blegwname}/count: query for events occurred in a specific BLE-GW	200	400
POST PUT	/table/users/entry: addition or modification of an administrator	201 200	400
DELETE GET	/table/users/entry/indexes/id/{id}: deletion or detail query for a specific administrator	204 200	400
POST PUT	/table/pushusers/entry: addition or modification for a push user	201 200	400
DELETE GET	/table/pushusers/entry/indexes/phonenummer/{phonenummer}: deletion or detail query for a specific push user	204 200	400
POST	/table/beacon/entry: BLE-GW's periodic notification with configuration information of beacons	201	400
POST	/group/pushresult: Acknowledgement of push reception from a smart device	201	400

The modification API for a specific beacon's configuration information is delivered to the BLE-GW from BLE beacon manager, and other APIs are delivered from BLE-GW or management terminal to BLE beacon manager. To periodically notify configuration information of beacons to manager, BLE-GW utilizes RBNM-API with HTTP POST and “/table/beacon/entry/” type URL. In the notification message, the configuration information of each beacon includes beacon name, MAC address, battery level, signal strength, UUID, and status values. The JSON based API for notification is as follows.

```
{ "blegwEntry": {
  "gwName": "blegw1",
  "ipAddress": "220.68.xx.xx",
```

```

"location": "anungnlab",
"numofBeacons": "3",
"beaconEntries": {
    "beaconEntry": {
        "beaconName": "reco",
        "mac": "D1:6D:7B:E3:B9:F7",
        "battaryLevel": "56",
        "rssi": "67",
        "uuid": "11111111-1111-1111-111111111111",
        "major": "2",
        "minor": "413",
        "status": "normal"}
    }
}
    
```

4. Implementation of BLE Beacon Network Management System

For processing of management information, BLE beacon manager exchanges RBNM-API defined as JSON based RESTful web service, with management terminals or BLE-GWs. BLE beacon manager is web service server of RBNM-API for web and hybrid app based GUI, and notification of BLE-GW. On the other hand, BLE-GW becomes web service server of RBNM-API for change and query of specific BLE beacon's configuration information. The implementation structure of BLE-GW and BLE beacon manager is shown in Figure 2.

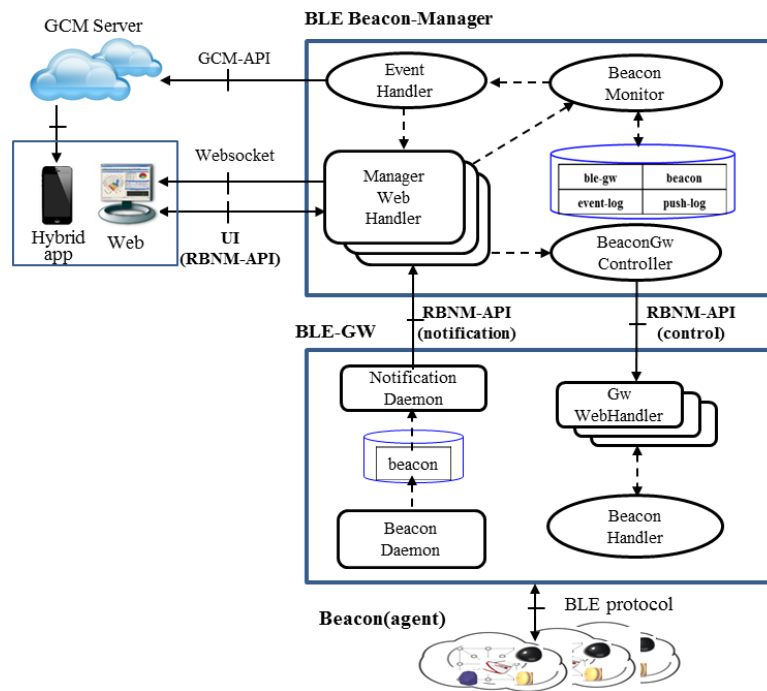


Figure 2. Configuration of BLE Beacon Network Manager and Gateway

4.1. Manager for BLE Beacon Network

BLE beacon manager performs account management of administrator, BLE-GW management, beacon management, and beacon event and fault management. Maria D/B was used to manage event log and configuration information of administrators, push users, BLE-GWs, and beacons. ManagerWebHandler processes RBNM-APIs for periodic

notification of beacon network information from BLE-GW and user interface of web and hybrid app based management. Jersey of JAX-RS reference implementation [18] and Apache-Tomcat were used to implement ManagerWebHandler where a new thread is created every RBNM-API request. BeaconGwController has the role of transmitting the request of RBNM-API to BLE-GW when beacon configuration information is set or changed. BeaconMonitor checks the status of each beacon and saves it into the beacon configuration table when beacon information is notified from BLE-GW.

If events of beacon to notify are detected, EventHandler inquires push user table and decides web browser and smart devices that events are to be notified to. EventHandler delivers events to the WebHandler of manager, and WebHandler transfers events to the web browser of administrator through Websocket's connection. When EventHandler notifies events to the smart devices, device token values of smart devices, push content, and manager key value are included in GCM3.0-API that is delivered to GCM server. In the response of GCM3.0-API as a result of push in GCM server, the number of messages sent to smart devices and the number of successful and failed push results are included. Failed result is returned if push service of a smart device is not registered in GCM server or application is deleted. If push is successful, message-ID is returned.

Entry of EventLogTable with event log is comprised of grade (critical, major, informational, and warning), cause, time, and beaconMAC. PushLogTable saves push results of events that are delivered to web browser and smart device. Entry of PushLogTable is comprised of message, pushCount, successCount, failureCount, and time. Message is a string of characters that are pushed, and pushCount indicates the number of smart devices to which push-server sends the push at the same time. SuccessCount and failureCount indicate the number of successful and failed results in pushing the events.

4.2. Gateway for BLE Beacon Network

The laboratory model of the BLE-GW that operates as an Android application was developed using Galaxy Tab (Galaxy Tab A with S pen Wi-Fi). BeaconDaemon thread extracts beacon configuration information (UUID, Major, Minor, battery level, etc.) from the advertising packets periodically transmitted from BLE beacons, and stores configuration information in the local database of the smart device. NotificationDaemon uses RBNM-API to periodically transmit beacon information stored by BeaconDaemon to BLE beacon manager. GwWebHandler calls BeaconHandler that performs interface with beacons when modification RBNM-API for a specific beacon's configuration information is received from BLE beacon manager. BeaconHandler creates a connection with the corresponding beacon and modifies beacon's configuration information.

5. Validation and Analysis of BLE Beacon Network Management

As shown in Figure 3, experimental environment composed of BLE-beacon manager and BLE-GW (RESTClient and JMeter) was established for verification and performance analysis of RBNM-API.

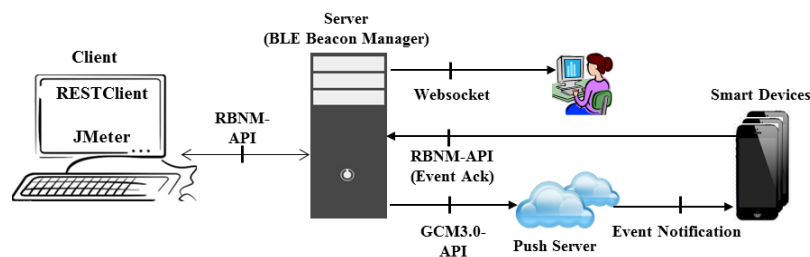


Figure 3. Test Environment of BLE Beacon Network Management

BLE-beacon manager has 4 GB of internal memory, 3.30 GHz CPU (dual core), and Windows 7 64-bit operating system. RBNM-API's client simulator was established in efficacy verification tool, RESTClient [3] to perform the verification. RESTClient can create various normal and abnormal request messages and check reply messages. JMeter [19] that emulates BLE-GW function supports various protocols such as HTTP, HTTPS, FTP, and SOAP, and it is a performance test tool of web program that is able to simultaneously transmit large-scale requests.

5.1. Validation of RBNM-API

BLE beacon manager uses regular expressions and JsonParser of JsonFactory to verify the validity of JSON based POST or PUT request message of RBNM-API, and verifies the validity of URL and arguments in GET or DELETE request message. Prewritten JSON message is transferred to BLE beacon manager when POST or PUT request type is initiated through RESTClient. We confirmed the validity verification and execution result of the API performed in BLE beacon manager by checking out the status line information of HTTP reply message.

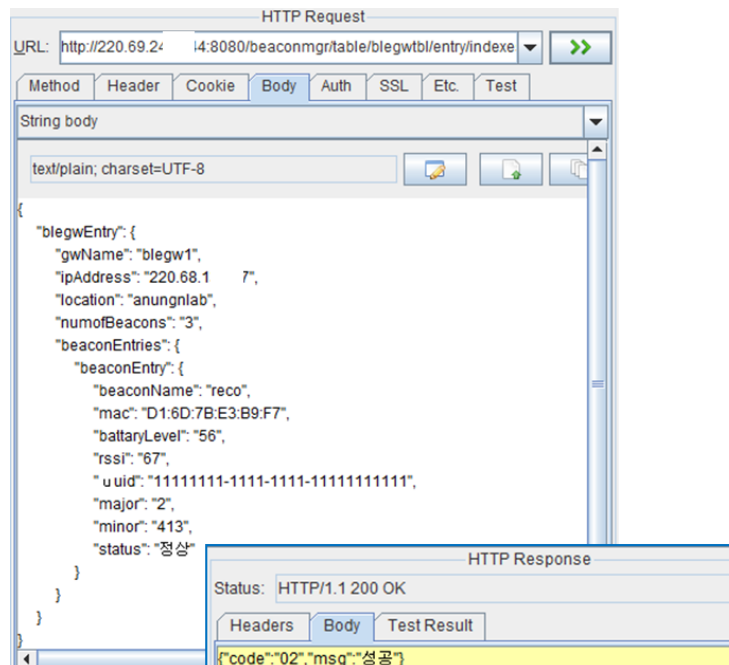


Figure 4. Verification Result of RBNM-API

Figure 4 shows the verification result of RBNM-API that is used by BLE-GW to periodically notify beacon configuration information to manager. When the verification and execution of the notification message included in the body of POST type message are successful, BLE beacon manager returns the reply message with “201 Created” in the status line. When the verification or execution of the notification message fails, BLE beacon manager returns “400 Bad Request” in the status line and strings with failure cause and location in the body of reply message.

5.2. Performance Analysis of RBNM-API

Using Apache JMeter [19], processing time in BLE beacon manager was measured according to the types of JSON parsers for handling RBNM-API and D/B connection methods for processing of beacon information periodically notified by BLE-GW. Processing time in BLE beacon manager excluding client and network latency was

measured and processing time was defined as the difference between the reception time of RBNM-API request and transmission time of reply. The number of threads for ManagerWebServer in the BLE beacon manager was set as 250 which is the default configuration of Apache Tomcat and for large-scale beacon network management, measurement was done for cases where up to 100 notification requests from BLE-GW were transmitted per second.

The notification message transmitted to the manager by BLE-GW has one or more beacon configuration information. Sequential and concurrent access mechanisms are possible to store notification message's beacon configuration information to database. Sequential access mechanism is a simple method that brings D/B connection to store single beacon configuration information in database and returns D/B connection, and then uses new D/B connection for the next beacon. Concurrent access mechanism is a method that brings D/B connection to store all configuration information of beacons included in the notification message and returns D/B connection.

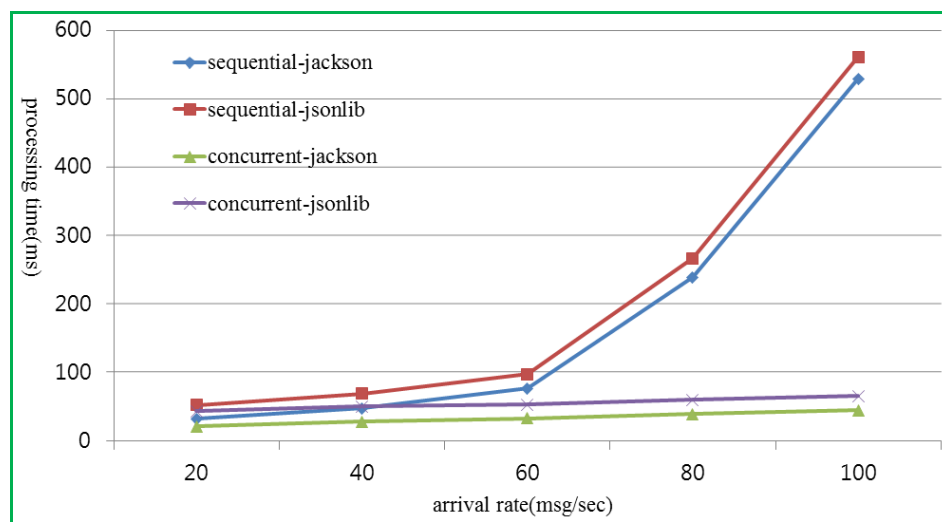


Figure 5. Processing Time according to Database Access Methods and JSON Parser Types

Figure 5 represents the processing time of BLE beacon manager according to JSON parser types (Jackson, JSonLib) for handling RBNM-APIs and D/B access methods (sequential, concurrent) for processing of beacon information. The number of beacon's configuration information in the notification message is 10 and DBCP (DataBase Connection Pool) has 15 connection pools. In the environment of Jackson parser, the processing time of sequential mechanism in message arrival rate 100 was 528.1ms and the processing time of the concurrent mechanism was 44.9ms. While in the concurrent access mechanism, there is gradual increase of processing time with the arrival rate, there is sharp increase of processing time according to the arrival rate in sequential access mechanism. For concurrent access method the processing time of Jackson type is 31.6% superior compared to that of JSonLib type in arrival rate 100.

5.3. Performance Analysis of Event Notification through Push Service

Websocket, Polling, and Comet techniques were compared in terms of stand-by time for notification of events. Polling technique is divided into Polling-A (10 seconds of polling cycle) and Polling-B (20 seconds of polling cycle) depending on the cycle of request. 500 events were designed to randomly occur between 0 and 10 seconds. Figure 6 indicates stand-by time for notification of events from Websocket, Comet, and Polling techniques. Stand-by time for event notification is defined as time from when event is

produced by manager to when it is delivered to web browser. Stand-by time of polling technique increased in proportion with the length of requested cycle, while the one of the Comet technique was estimated to be about 2.2 seconds, and the one of WebSocket was measured to be near 0 second.

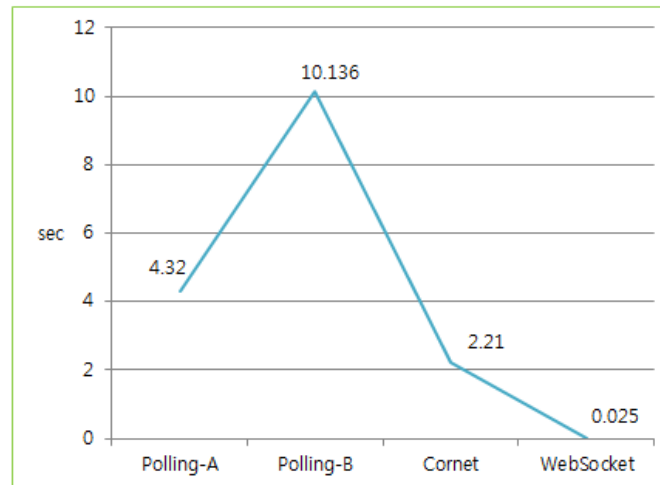


Figure 6. Waiting Time of Event Notification to Web Browser

Even if events are lost between GCM server and smart device, they are successfully indicated on the responding message of GCM3.0-API that is delivered by GCM server to the manager. If a smart device is turned off, or if there is an error on network, GCM server preserves a message up to 4 weeks and push events if it is feasible for smart devices to accept them. It is possible to confirm whether event messages are delivered from manager to GCM server of Google with GCM3.0-API that is returned from GCM server. However, it is not possible to confirm whether events are delivered from GCM server to smart devices on a real-time basis. As for important event, smart device uses RBNM-API confirming the receipt of event from a smart device to a manager. The confirming RBNM-API uses POST method and “http://beacon-manager-address/group/pushresult” type URL.

Delay in notification is defined from the time when GCM3.0-API request is sent from manager to GCM server to the time when the receipt of event from the smart device is confirmed by RBNM-API. Using 5 Android smart phones and 4 iOS smart phones in the same area, delays in notification have been measured. Average delay in notification of 5 Android phones turned out to be 590ms, and the one of 4 iPhones was 3,230ms. Since iPhone provides push service through GCM server from Google instead of APNS server, delay in notification of events of iPhone turned out to be higher than the one of Android phone. It was confirmed that there have been significant differences in delays between maximum and minimum values of notifications depending on the status of network.

6. Conclusion

Following the expansion of services using BLE beacon and market size, this study suggested management configuration diagram and management function of BLE beacon network that consists of beacons, gateways, and manager. JSON based RESTful web service, RBNM-API was defined for exchange of beacon network management information, and using RESTClient, the validity of the defined RBNM-API was verified. A manager operating on a PC with Windows 7 and a gateway operating on a Galaxy Tab were developed as laboratory models to test the management of BLE beacon network. BLE beacon manager’s processing time was

measured according to JSON parser types and database access methods for handling of beacon configuration information. For implementation of large-scale beacon network management system, Jackson parser and concurrent access mechanism are required for superior processing time. In the environment of concurrent access mechanism, processing time of beacon configuration information gradually increased with arrival rate. We compared Websocket, web's push service technology with existing Polling and Comet technologies in terms of waiting time of event notification. Polling technique turned out that stand-by time of a notification has been increased when a cycle of request increased. However, since stand-by time of Websocket was closer to 0 second, it was feasible to minimize overhead of network and achieve notification in real-time. We also tested event notifications toward smart devices and measured notification delays with GCM3.0-API and RBNM-API. Since iPhone provides push service through GCM server from Google instead of APNS server, delay in notification of events of iPhone turned out to be higher than the one of Android phone.

Acknowledgments

This work was supported by a grant from 2015 Research Fund (Research Professor) of Andong National University. This paper is a revised and expanded version of a paper entitled "BLE Beacon Network Management based on Manager and Gateway", presented at MITA2016.

References

- [1] T.Y. Ha, S.W. Cho, J.H. Park and J.M. Jeung, "Analysis of BLE Beacon Device's Industrial Trend and Possible Problem", Korean Society For Internet Information Conference, (2015), pp. 199-200.
- [2] Y.M. Park, A.K. Moon, H.K. Yoo, Y.C. Jung and S.K. Kim, "SOAP-based Web Services vs. RESTful Web Services", Electronics and Telecommunications Trends, vol. 25, no. 2, (2010), pp. 112-120.
- [3] S. Chandran, "WizTools.org RESTClient: A Quick & Dirty Guide", WizTools.org, (2013).
- [4] W.R. Park, S.C. Park and D.Y. Kim, "Design of Message Push System Using WebSocket Based on Hybrid App", Proceeding of the Spring Conference of the Korean Society for Internet Information, (2014), pp. 189-190.
- [5] Google Cloud Messaging, <https://developers.google.com/cloud-messaging> (accessed June, 9, 2016), (2016).
- [6] A. Pras, T. Drevers, R. van de Meent and D. Quartel, "Comparing the Performance of SNMP and Web Services-Based Management", IEEE Transactions on Network and Service Management, vol. 1, no. 2, (2004), pp. 72-82.
- [7] A. P. Castellani, M. Gheda, N. Bui, M. Rossi and M. Zorzi, "Web Services for the Internet of Things through CoAP and EXI", IEEE International Conference on Communications Workshops (ICC), (2011), pp. 1-6.
- [8] Mobile push and nPush, <http://d2.naver.com/helloworld/1119> (accessed July, 2, 2016), (2011).
- [9] N. Li, Y. Du and G. Chen, "Survey of Cloud Messaging Push Notification Service", International Conference on Information Science and Cloud Computing Companion, (2013), pp. 273-279.
- [10] S.H. Park, Y.C. Oh and C.B. Kim, "Image Security System using Push Server and Smart Device", Journal of Korean Navigation Institute, vol. 8, no. 6, (2014), pp. 588-593.
- [11] J.A. Kim and S.I. Kim, "Beacon's User Analysis and Proposal Guidelines on IoT Environment", Journal of Communication Design, vol. 54, (2016), pp. 213-223.
- [12] S.J. Lee, "Beacon base Service Offline Market Activate Expectations", KISTI Market Report, vol. 4, Issue12, (2014).
- [13] J.W. Son, K.R. Cho and B.K. Jang, "Beacon-management Scheme using the Moving Route", Korean Institute of Information Scientists and Engineers Conference, (2015), pp. 379-380.
- [14] Cisco's BLE Management Platform, http://www.cisco.com/c/en/us/td/docs/wireless/mse/release/notes/cmx_10_1_rn.pdf (accessed July, 9, 2016).
- [15] S.B. Han and J.H. Song, "A Push Technique of the Product Information by Beacon's MacAddress", Journal of the Society of Convergence Knowledge, vol. 4, no. 1, (2016), pp. 13-21, 2016.
- [16] J.A. Lee, "Emergence of Beacon Services and Diffusion of New Businesses", IT & Future Strategy Report, no. 8, (2014).

- [17] I.S. Choi, C.H. Kim and Y.W. Cha, "Design and Implementation of Integrated Management System for Gas Station using Web Service and Sensor Network", Journal of KIIT, vol. 10, no. 9, (2012), pp. 105-115.
- [18] Jersey 2.0, GlassFish Community, <http://jersey.java.net> (accessed March, 6, 2016), (2016).
- [19] Apache JMeter, Apache Software Foundation, <http://jmeter.apache.org/> (accessed Marched, 6, 2016).

Authors



YoungWook Cha, he obtained his PhD in Computer Engineering from KyungPook National University in South Korea in 1998. He worked for ETRI as a senior researcher and he is currently a professor of Computer Engineering Department at Andong National University. His research interests include network management, NGN, security, and network QoS control.



DongMin Kim, he is now taking a B.S. in Computer Engineering at the Department of Andong National University in South Korea. His research interests include network management, RESTful web service, and information modeling.



ChoonHee Kim, she obtained her PhD in Computer Engineering from KyungPook National University in South Korea in 2000. She worked for ETRI as a researcher and she is currently a professor of Electronic & Information Communication Engineering Department at Daegu Cyber University. Her research interests include network management, and network QoS control.