The Design of Integrated Mobile SNS Gateway Structure

Shinho Lee, Insik Jung, Hyeonwoo Kim, Hongtaek Ju
Dept. of Computer Engineering
Keimyung University
Daegu, Republic of Korea
{ leeshinho, dlstlrdl77, hwkim84, juht }@kmu.ac.kr

***Abstract*—Lately, as usage of mobile end terminals such as smart phones has increased, mobile traffic has also increased rapidly. This paper proposes the design of the Integrated SNS gateway server as a method to solve the SNS traffic that rapidly grew. At present, mobile SNS applications renew information by approaching and responding to each single corresponding SNS server. The proposed Integrated SNS Gateway combines these SNS servers into one and sets a gateway server in between the client and the SNS server, so that traffic for repeated data request is decreased, and provides a method to improved mobile communication environment.**

Keywords—MQTT; SNS; Gateway; Squid Cache Server; Push Notification Service;

I. INTRODUCTION

As usage of mobile devices such as smart phone rapidly increased and became common, mobile traffic also increased sharply. Of the mobile traffics, SNS traffic especially shows a very fast-growing trend [1], [2]. Currently SNS utilizes push notifications in mobile environment, and Y. J. Lee and three others proposed an improvement to push notifications like Abstract Push Framework [3]. SNS applications that use such type of push notifications have different information provided and characteristics for each SNS, so the users utilize diverse types of SNS. These SNS applications approach each SNS server to offer information to the user. For example, the most popular SNS, Facebook and Twitter, use individual SNS application for each SNS server that provides service to renew information.

Such 1:1 communication method brought wanted information by allowing the user to connect to each web server or FTP server, like in early web or FTP, but as the web became enormous, troubles like complicated communication and increased traffic arose. To respond to this, J. Y. Kim and two others used a method [4] that utilizes web caching to eliminate bottleneck phenomenon, which places caching device that resolved communication traffic and improved problems with quick response. If mobile SNS approach this method so that there is an integrated SNS gateway server in the middle and utilize cache and synchronization, mobile SNS traffic will decrease and communication speed will increase.

This paper suggests Integrated SNS Gateway top structure design and communication method. That will decrease mobile SNS traffic and increase communication speed through integrating SNS through cache and synchronization. The suggested Integrated SNS gateway communicates through two steps. In between the gateway and each of the SNS servers, the existing SNS self-protocol is used for communicating, and in between the mobile device and the gateway, newly designed Integrated SNS Communication protocol is used for communicating. Integrated SNS Application combines all functions that existing SNS applications provided. Although Integrated SNS Application has to be installed separately and the services provided by each SNS could be different, positive effect is made because of the improvement of communication performance. The key elements of Integrated SNS Gateway are the synchronization with cache, integrated certification function, and the adaptor for communication with each SNS server, and Integrated SNS Gateway provides a method to connect them in an organized method.

II. RELATED WORK

Generally, SNS communication method is divided into two types: push notification method and contents load method. Push notification is a communication method that keeps the communication to not break and sends messages to a certain device when an event happens in the server. Push notification is a method that can increase efficiency in battery and traffic better than the polling method, which the devices confirm with the server at all times. For the push notification method, we will utilize the Message Queuing Telemetry Transport (MQTT), which is an open protocol presented by International Business Machines Corporation (IBM).

MQTT is suitable protocol for implicating the Integrated SNS Gateway in a different Operating System (OS) environment or SNS protocol. MQTT has merits in that it can implement push notification service individually without depending on a vendor service company and in that there is no limit in messaging. Fig. 1 explains the message transmission progress of MQTT. MQTT consists fundamentally of 1 Broker Server, 2 Publish, and Subscribe client [5]. The Broker server acts as an intermediary of message delivery of a topic between 2 Publish and Subscribe clients. When the Publish client issues

This research was financially supported by the Ministry of Education, Science Technology (MEST) and National Research Foundation of Korea (NRF) through the Human Resource Training Project for Regional Innovation (2012H1B8A2025942), and NIA(National Information Society Agency).

the Topic and delivers the message to the Broker server, Subscribe client can subscribe to the Topic of interest.

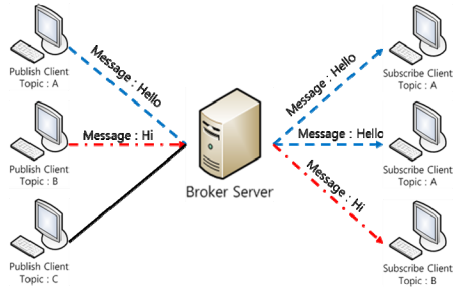

Fig. 1. MQTT Message Transmission Progress

Contents Load is a method that when such push notifications notifies the user, the device requests needed contents to a server and gets a response.

To implement such Contents Load method, we use JavaScript Object Notation (JSON). JSON is a lightweight data transfer format, which has a free data type and is easy to read and write, and is able to express data format more simply than can XML [6]. Because it has a more comfortable structure than that of XML, parsing is faster and easier, and because it is useful in mobile environment or server, JSON is more efficient than XML[7].

III. The Design of Integrated Mobile SNS Gateway Structure

A. *Integrated Mobile SNS Gateway*

Integrated Mobile SNS Gateway is placed between the Integrated Mobile SNS Application, installed on a user's device, and the existing SNS server. When each of the users utilizes the Integrated Mobile SNS Application to set the SNS being used, then the gateway provides information that the user wants by using the set SNS. Fig. 2 is a comparison of the method that communicates from a general SNS application to each of the SNS servers and the method that uses gateway.

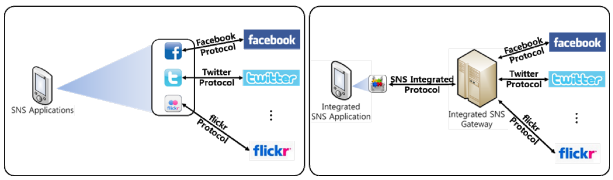

Fig. 2. Comparison of Existing Method and the Gateway Method

The gateway that this paper proposes saves the information of the user who uses the Integrated SNS Application, provides or saves data by receiving requests from the Integrated SNS Application, and renews and brings the user's SNS information and new information, then provides them to the Integrated SNS Application. The Integrated SNS Gateway places a gateway server in between the device and the SNS server to cache and synchronize frequently requested contents such as pictures, videos, friends list, or contacts so that it can provide specific contents from the gateway's cache engine or synchronization engine when requested such contents without using the SNS server. As caching device, a gateway is placed in the SNS server and the device for these contents, traffic load for the same contents requests between SNS server and multiple devices. Our proposed caching and synchronization method shows increased efficiency in traffic and transmission performance than requesting SNS server for contents at all times.

B. *The design of Integrated Mobile SNS Gateway Structure*

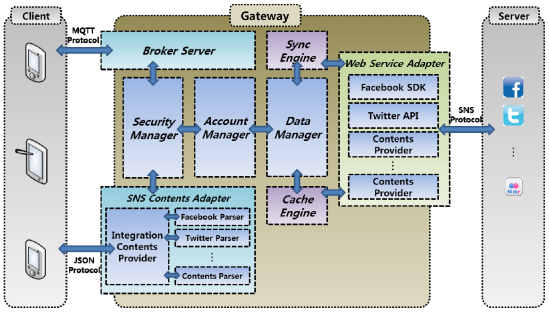

Fig. 3. Structure of the Integrated SNS Gateway

Fig. 3 is the server structure of the Integrated SNS Gateway. For the Integrated SNS Gateway, each SNS server has to be connected through the corresponding self-protocol for each SNS. In Gateway, web service adapter module does this role, and we use SDK or API provided by each SNS for web service adaptors to transmit and receive each SNS contents. Also, to save and provide information of the Integrated SNS application users, SNS gateway's Security Manager, Account Manager, and Data Manager performs the security, account, and data management.

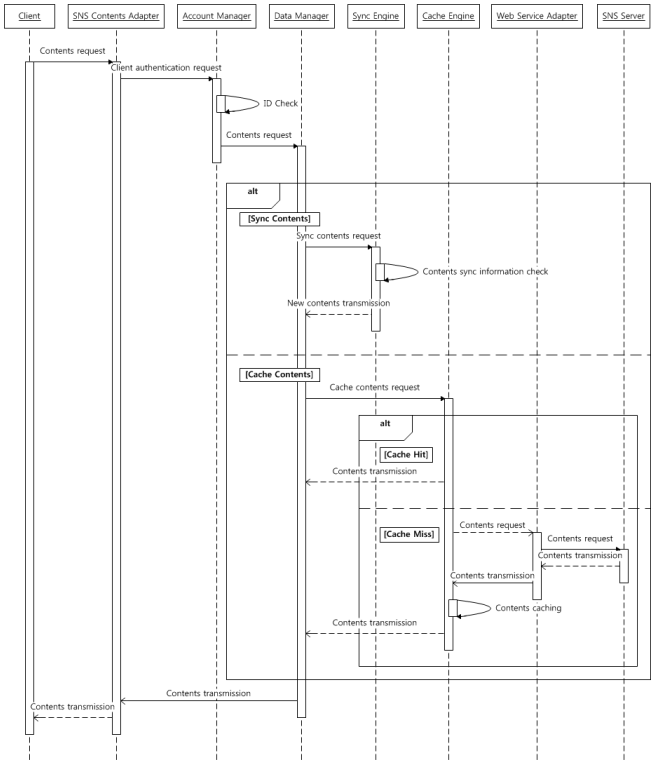

Fig. 4. SNS Contents Load Sequential Diagram

Fig. 4 illustrates the requesting and processing of the contents. The cache engine module of the Integrated SNS Gateway uses the Squid cache server, which is an open source project. The standard protocol SyncML is utilized for the synchronization engine module. We use another open source project Mosquitto as an intermediary role in for the push notifications in gateway. When the corresponding push notification is transmitted from the SNS server to the gateway's web service adapter module, the Broker server module mediates and transmits to the client.

C. Integrated SNS Protocol Design

1) Push Notification Protocol Design

MQTT uses Topic for the push service between the Publish/Subscribe client and the Broker server. MQTT provides Topictree to allow subdivision of the Topic, and because in integrated SNS each user uses different SNS that have different functions, push notification method has to be managed in subdivisions for each SNS. In order to discern push notification service from module in functional order, definition of the Topictree was set as following Fig. 5.

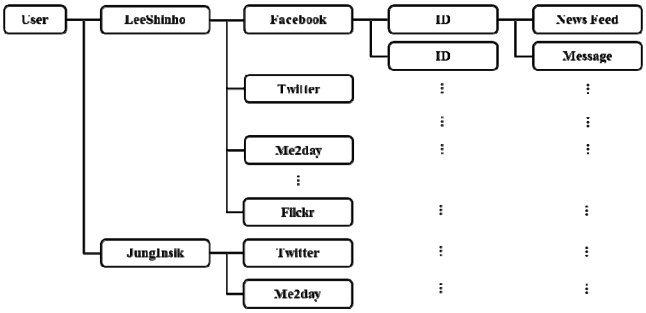

Fig. 5. Example of Topictree for Push Notifications

Fundamentally, Topictree is distinguished based on user ID, type of SNS, and the function of SNS. We designed a new Integrated SNS Gateway protocol utilizing MQTT's payload part so that push notification service for each SNS can be distinguished. Fig. 6 is a result of protocol design for push notification service for the Integrated SNS Gateway.

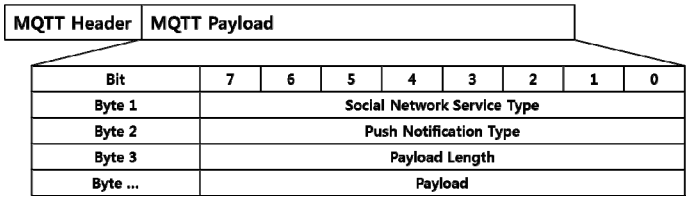

Fig. 6. Push Notification Protocol Design using MQTT

We will analyze push notification by each SNS and functions based on the above design so that integrated SNS application can provide push notification service for each SNS and functions.

2) Contents Load Protocol Design

When the user receives a certain push notification, contents are requested in order to browse the received push information. In order to take care of such contents request in the Integrated SNS Gateway, we used JSON. JSON is fundamentally structured out of Key and Value. Fig. 7 is the result of designing SNS protocol for contents request and response.

```
Contents Request
{
        "ID" : "Client ID",
        "SNS Type" : "Request SNS Type"
        "Message Type" : "Request"
        "URL" : "Request URL"
}

Contents Response
{
        "ID" : "Client ID",
        "SNS Type" : "Response SNS Type"
        "Message Type" : "Response"
        "Contents Name" : "Response Contents Name"
        "Contents Binary" : "Response Contents Binary Data"
}
```

Fig. 7. Contents Load Protocol Design Using JSON

IV. CONCLUSION

In this paper, we proposed a method to improve communication performance using Integrated SNS Gateway in order to respond to an explosively growing mobile SNS traffic. In order to execute the proposed method, the final top design of the Integrated SNS Gateway, the method of communication between the mobile device and gateway SNS server, and the resulting design for communication protocol were introduced. The performance of the most important factors, cache function and the synchronization function, will be measured through later research so that it can be reflected in implementing the gateway. Our plan is to precede implementation based on mobile SNS gateway that is designed through later research and to measure the efficiency in mobile SNS traffic through the mobile SNS gateway

REFERENCES

[1] Y. R. Choi, J. Y. Chung, B. C. Park, and J. W. K. Hong, "A study on system architecture for application-level mobile traffic monitoring and analysis," KNOM Review, vol. 14, no. 2, pp. 10-21, Dec. 2011.

[2] H. Min, M. S. Kim, "Towards Smart Phone Traffic Classification," Network Operations and Management Symposium (APNOMS), 2012 14th Asia-Pacific, pp. 1-4, Seoul, Korea, Sep. 2012.

[3] Y. J. Lee, J. S. Oh, B. G. Lee, "Logical Push Framework for Real-time SNS Processing," Computational Aspects of Social Networks (CASoN), 2012 Fourth International Conference on, pp. 47-51, São Carlos, Brazil, Nov. 2012.

[4] J. Y. Kim, K. W. Cho, K. Koh, "A proxy server structure and its cache consistency mechanism at the network bottleneck," Computer Software and Applications Conference, 1999. COMPSAC '99. Proceedings. The Twenty-Third Annual International, pp. 278-283, Phoenix, AZ, USA, Oct. 1999.

[5] P. Th. Eugster, P. A. Felber, R. Guerraoui, and A. Kermarrec. "The many faces of publish/subscribe," Journal ACM Computing Surveys (CSUR), vol. 35, Issue 2, pp. 114-131, Jun. 2003.

[6] N. Nurseitov, M. Paulson, R. Reynolds, and C. Izurieta "Comparison of JSON and XML data interchange formats: A case study," in Proc. ISCA 22nd Int. Conf. Computer Applicat. Ind. Eng. (CAINE 2009), pp 157-162, San Francisco, California, USA, Nov. 2009

[7] C. Rodrigues, J. Afonso and P. Tomé, "Mobile Application Webservice Performance Analysis: Restful Services with JSON and XML," ENTERprise Information Systems Communications in Computer and Information Science Volume 220, pp. 162-169, Oct, 2011.