# Person Wide Web: Active Location based Web Service Architecture using Wireless Infrastructure

Seungjae Shin*, Pyung Kim*, Yeojeong Yoon*, Seongbea Eun†, and Hyunsoo Yoon*
*Dept. of Computer Science, Korea Advacned Institute of Science and Technology, Daejeon, Rep. of Korea
†Dept. of Information and Communication Engineering, Hannam University, Daejeon, Rep. of Korea
Email: {sjshin, pkim, yjyoon}@nslab.kaist.ac.kr, sbeun@hnu.kr, hyoon@nslab.kaist.ac.kr

***Abstract*—Due to the recent advancements of Wi-Fi technologies, people can receive various types of data services via their portable devices such as laptop, smart-phone, and PDA. Especially, as the localization technologies are well developed, the LBS (Location Based Service) is recognized as a promising technology in the upcoming ubiquitous era. In most of LBS applications, users generally tend to visit shops, restaurant and public facilities in his surroundings. To exploit this feature, we propose Person Wide Web (PWW), a new type of location based web service architecture that effectively recognizes the location specific web resources to the mobile device user based on his geographical position. Among so-many resources in the WWW, only the things closely located to the user, make up his PWW. By using the PWW, various location specific data services such as shop advertisement, restaurant menu, and discount coupon, are expected to form a new type of business model in which marketers can actively promote the mobile device users to gladly use their services. We implemented a prototype of PWW system on Android platform and saw that our prototype has reasonable performance on service establishment and message delivery time. We believe the PWW has a potential to open up a new market area of provider triggered wireless data services.**

***Index Terms*—LBS (Location based Service), Wireless Web, Location Specific Service, Ubiquitous Service, Application Protocol**

## I. Introduction

The improvements of Wi-Fi technology allow people to easily access the Internet via lap-top PC and smart-phone in a public place [1]. In addition, cellular technology is also evolving to provide not only voice calls, but also large scale data contents [2], [3]. Hence, currently the difference in purpose between telecom and datacom technologies is getting more blurred so that there are various data services targeting wireless environments [4], [5], [6]. Especially, LBS (Location Based Service) which is the data service utilizing location information of various objects, is emerging as a promising technology in a ubiquitous environment. However, when we observe the widely used LBSs such as map search [7], navigation service [8], most of them has different system architecture from each other. Furthermore, users have to use different client software for each of them.

For above reasons, several works have been introduced in literature to operate LBS application on web based environments which are familiar with the users [9], [10], [11]. Web based LBS does not require any specific client SW, because the user can use it in the same way when they use normal web sites. Therefore, all the LBS providers are only required to establish their web sites. Moreover, the user does not have to install client program for each LBS and to learn how to use them. Even though such advantages, there exists an important challenge to be coped with for the success of web based LBS. That is how the users recognize web resources which are related to the objects closely located with their current location. (From now on, we will refer to such web resource as *geographically effective* web resource.) However, it is bothersome and difficult works for each LBS user to recognize his *geographically effective* web resources. By this reason, we introduce Person Wide Web (PWW), a new type of web service architecture, which automatically recognizes *geographically effective* web resources and notifies them to the users in a periodic manner. PWW can be understood as a subset of WWW, because it is a selection of *geographically effective* web sites. In contrast to the existing web service technologies passively waiting until the user requests the service, PWW tries to trigger intended requests of service from users by actively delivering PUSH messages that includes abbreviated information for location-specific web resources. PWW has a potential to develop a new service model emphasizing the active promotion of marketers and providers.

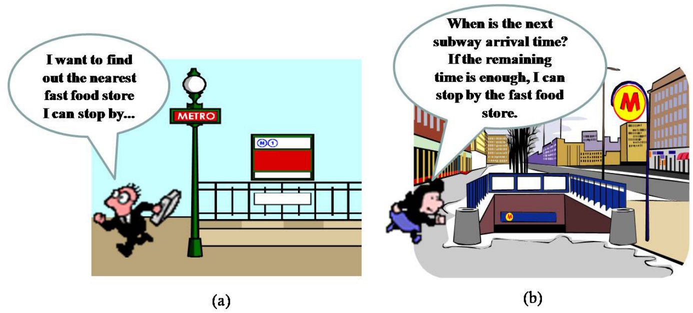


Fig. 1. Two Examples: The needs for the PWW

Let see the situational examples motivating the needs for PWW. In Fig. 1 (a), Albert is on his way to the office of his business partner. And, he got off the subway just now. Due to the very busy schedule, he skipped his lunch. Thus, he wants to stop by a fast-food store around the subway station. But, he has to go around the street or ask someone to find out where the fast-food store is, because he is not accustomed with his current surroundings. On the other hands, in PWW, an active information announcer, called PUSH server, lets Albert know the integrated information such as menus, prices, rough maps, and electronic discount coupons of each fast-food store in the surroundings. Thus, by using PWW client on his smart-phone, Albert easily get his *geographically effective* web resources about the fast-food stores in his neighborhood at once. Then, he can just choose his most preferred one and stop by there. If he does not want this type of service, all the things need to do is just not executing PWW client on his smart-phone.

Next, let's see another example in Fig. 1 (b). In this example, Betty is planning to get on the subway as soon as possible to go to her workspace. Similar to Albert, she also skipped her lunch due to the busy schedule. She wants to stop by fast-food store if the remaining time until next subway arrival is enough. At this moment, PWW system delivers the next subway arrival time to the client program on her smart-phone. Betty can determine whether to stop by fast-food store or just going to platform according to remaining time.

The above examples are just small part of the needs for the PWW. Regarding the activeness and location-specific nature of PWW, we can figure out so-many service scenarios appealing mobile device users. In short, PWW has the potential to create a large market area.

In this paper, we specify the PWW as a general framework to be used for various location-specific services and present our prototype as the implementation example. The remaining part of this paper is as follows: In section II, we explain the overall architecture of the PWW and how PWW system behaves on the specific service scenarios described in Fig. 1. The implementation result and performance evaluations are present in Section III. Finally, we make our conclusion and suggest future works for PWW in Section IV.

## II. PWW Architecture

In this section, we explain the PWW as an infra-system supporting various location based data services. And, we will describe how the protocols that make up PWW behave by using two service scenario examples.

### A. Basic Framework

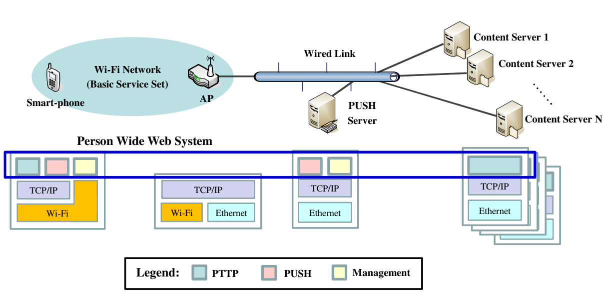


Fig. 2. Person Wide Web System

Fig. 2 shows the PWW system and its component protocols. As depicted in Fig. 2, the PWW system runs over the TCP/IP and Wi-Fi stacks. It consists of following three sub-protocols:

1) **PTTP (PWW Text Transfer Protocol):** deliver web documents for various data services to the mobile device users. It is very similar to HTTP [12] and WAP [5].
2) **PUSH Sub-Protocol:** periodically send PUSH messages including *geographically effective* web resources.
3) **Management Sub-Protocol:** manage the connection between the PUSH server and the user's mobile device.

PTTP is basically an extension of HTTP. In PTTP, URL (Universal Resource Locator) [13] should represent not only a logical web address, but also the physical location of the object related with the web resource. The PWW mainly focuses on the location based data services in which the physical location of the object largely affects the user's decision whether to willingly use the service or not. Hence, it is desirable to make the mobile device user to infer the physical location of objects based on its URL. However, it is difficult to do such that because traditional URL uses IP address or domain name as a host identifier in the network. Thus, we need to use a new type of host identifier also implying geographic location of the object. From now on, we refer to such host identifier as LID (Location Identifier). In our design, as observing that fixed-line phone number assignment is generally based on the geographical location, we use fixed-line phone number as LID. In fact, the LID can be defined in different ways. For example, the combination of zip code and street address also can be used as LID. We believe that the standardization of how make LID is necessary for the success of PWW. But, we do not discuss that issue here because the standardization is out of the scope of this paper. By using LID in URL, the user can identify geographical proximity among different objects (ex: store, shop, restaurant, business office) based on the similarity of their URLs. One notable thing here is that PTTP certainly needs the directory service which resolves LID to IP address for the same reason as DNS (Domain Name Service) does [14]. But, considering the location specific nature of PWW, most PWW users only use the location based services in the near surroundings. Thus, we don't have to establish the large directory service like current DNS infrastructure. We just need the small-sized local directory service which only handles LID of resources for objects in the surroundings. We are planning to make the management sub-protocol to be responsible for such as task in the future.

```
GET /directory/page.html PTTP/1.0
Host: 82.42.627.8886
Connection: close
User-agent: Mozilla/4.0
Accept-language: en
```

(a) PTTP Request

```
PTTP 1.0 200 OK
Connection: close
Date: Sun, 21 Nov 2010 12:25:13 GMT
Server: Apache/2.0.0 (Unix)
Last-Modified: Sat, 11 Sep 2010 13:57:08 GMT

<html>
<body>
Welcome to Person Wide Web!!
</body>
</html>
```

(b) PTTP Response

Fig. 3. A Example of PTTP Messages

Fig. 3 shows an example of PTTP messages. Fig. 3 (a) is the PTTP request message to locate the URL, pttp://82.42.627.8886/directory/page.html. In this example, LID is "82.42.627.8886" made by using the fixed-line phone number. And, Fig. 3 (b) is an example of PTTP response including an HTML web document. We can see that the message format of PTTP is almost same as that of HTTP.

PUSH sub-protocol is a unique service announcement protocol of the PWW. It tells the mobile device user about *geographically effective* web resources which are related to objects (restaurants, stores, facilities, services and so on) near the user's current location. The body of a PUSH message is a XML web document made by the marketer. In other words, the content of a PUSH message is a web page including not only text but also multimedia data such as picture, figure and sound. Thus, the PUSH message carries out various types of contents, from service advertisement to special types of item like a business card, coupon. By this reason, PWW is more extensible and cost-effective than WAP [5] that just delivers the simple types of PUSH message by only using MMS (Multimedia Messaging Service).

After that, he will delete the PUSH message including the coupon from his client SW.

It is efficient to make a local server only have whole responsibility of this PUSHING, because PWW is used for location specific services, Hence, the delivery of PUSH message is done by the PUSH server connected with the BSS (Basic Service Set) [1] the user belongs to. This configuration improves the performance of PUSH sub-protocol such as message delivery time, bandwidth efficiency compared with when each PTTP web server individually does their own PUSHING. In this configuration, all each marketer is only required to create a XML web document for PUSH message, and then uploading it to a local PUSH server by using file transfer service like FTP.

This PUSH server periodically broadcasts the PUSH messages to the users in the BSS connected to it. Based on the contents included in messages, the broadcast interval can be set from few seconds to few minutes by the administrator. When the user receives same content multiple times due to the long residence time, the Client SW discards this kind of PUSH messages.


Fig. 4. A Example of PWW Client

```
< ?xml version="1.0" encoding="euc-en"?>
<PUSH-MESSAGE>
<TYPE-CODE>F-1</TYPE-CODE>
<TIME-STAMP>2010-04-28-16:44:25</TIME-STAMP>
<HASH-STRING>KB29DKFLHP10IPQW</HASH-STRING>
<BODY>
    <CONTENT>
    <![CDATA[
        < ?xml version="1.0" encoding="euc-en"?>
        < ?xml-stylesheet type="text/xsl" href="cdcatalog.xml" ?>
        <catalog>
            ••••••
        </catalog>
    ]]>
    </CONTENT>
    <URL>pttp://82.42.350.5569</URL>
</BODY>
</PUSH-MESSAGE>
```

Fig. 5. A Example of PWW PUSH Message

Now, let's look at Fig. 4 which shows the example of client software that receives PWW PUSH messages. Firstly, we can see that the frame titled Push Message List, displays the metadata (reception time, message title) of each received PUSH message. When the user clicks a row he wants, the content of that message is displayed in the frame titled Web Document Viewer. (Note that the content of a PUSH message is a XML web page.) In this example, the content of the PUSH message is an electronic discount coupon from a waffle house named *De Bruxelle*. As we see, the coupon includes a picture, rough map, and barcode for a check. If the user wants to know more about *De Bruxelle*, he can visit its PTTP web site at the URL "pttp://82.42.350.5569". By clicking GO button, the user can see the official homepage of *De Bruxelle* displayed in Web Document Viewer frame. When the user visits the *De Bruxelle* in person, he will open the client SW and then use the coupon.

PWW push message is a string of XML document type as described in Fig. 5. The functions of each field are described as follows:

- **TYPE-CODE** represents the type of content. For example, we can notate *F-2* as "French-food restaurant", and *F-3* as "Italian-food restaurant", and so on.
- **TIME-STAMP** is the time recorded by the PUSH server when the message is transmitted.
- **HASH-STRING** is used for checking whether if the message includes same content received before. It can be generated by the public hash algorithm such as MD5. This is more efficient than checking whole content of messages.
- **BODY** is the content of message. It consists of two subfields, content part and URL part. One notable thing here is that CDATA keyword is necessary because content part includes another XML web document.

```
< ?xml version="1.0" encoding="euc-en"?>
<PWW-MANAGEMENT-MESSAGE>
<MESSAGE-TYPE>CONNECT</MESSAGE-TYPE>
<USER-ID>SNOOPY0324</USER-ID>
<IP-ADDR>192.168.0.8</IP-ADDR>
<MAC-ADDR>00-E0-4D-F8-38-07-</MAC-ADDR>
<TYPE-OF-INTEREST>F02</TYPE-OF-INTEREST>
</PWW-MANAGEMET-MESSAGE>
```

(a) CONNECT Message

```
< ?xml version="1.0" encoding="euc-en"?>
<PWW-MANAGEMENT-MESSAGE>
<MESSAGE-TYPE>CLOSE</MESSAGE-TYPE>
<USER-ID>SNOOPY0324</USER-ID>
</PWW-MANAGEMET-MESSAGE>
```

(b) CLOSE Message

```
< ?xml version="1.0" encoding="euc-en"?>
<PWW-MANAGEMENT-MESSAGE>
<MESSAGE-TYPE>ALIVE</MESSAGE-TYPE>
<USER-ID>SNOOPY0324</USER-ID>
</PWW-MANAGEMET-MESSAGE>
```

(c) ALIVE Message

Fig. 6. Examples of PWW Management Message

Management sub-protocol is the signaling protocol to manage the connection between the PUSH server and the user device. Fig. 6 shows the examples of signaling messages in this protocol. Message (a) is CONNECT message transmitted by the user device to connect to the PUSH server. When the PUSH server receives a CONNECT message from a user device, it generates a per-connection state information which includes a user ID, IP address, MAC address and type of interest. Consequently, the PUSH server periodically broadcast PUSH messages to the users registered to this per-connection state list. Message (b) is CLOSE message for closing the connection to the PUSH server when a user ends PWW Client program or leaves the BSS. When the PUSH server receives this message from a user device, it deletes the connection state information of the user from its per-connection state list. Message (c) is ALIVE message which is used for notifying the PUSH server that the user is still connected to it. This is similar to ALIVE messages used in some kind of protocols [15], [16]. If a user do not send ALIVE message during a certain period, the PUSH server regard that the connection is closed and delete the information of the user from its per-connection state list.

### *B. Possible Service Scenarios*

In this section, we will illustrate how the sub-protocols, described in section II, behave in PWW systems. We firstly make two service scenarios based on the motivating examples in section I. Then, we will explain the detailed behaviors of sub-protocols for each scenario.

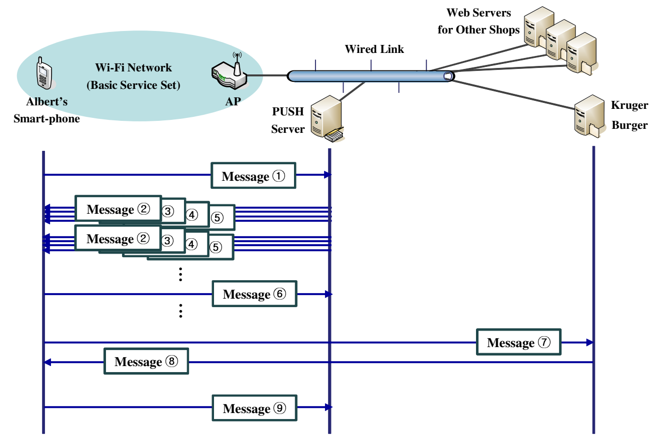


Fig. 7. PWW Scenario 1: Shop Advertisement Service

*1) Location Specific Shop Advertisement Service (Fig. 7):* In scenario 1, Albert just got off the subway and wants to stop by near fast-food store to buy something to eat. To get some information about that, he starts his PWW client program on his smart-phone. After that, the PWW system behaves as follows:

1) From Albert's device, CONNECT message (①) is delivered to the PUSH server. The PUSH server registers the connection to him in its per-connection state list.
2) The PUSH server periodically delivers its PUSH messages (②, ③, ④, ⑤) to Albert's client (ex: once every 60 seconds).
3) Now, Albert's PWW client stores PUSH messages (advertisement) about various shops such as *Kruger Burger*, *Starbucks*, *Philo Book Cafe*, *Tom's Drugstore* (②, ③, ④, ⑤). And, it also sends ALIVE message (⑥) in a periodic manner (ex: once every 30 seconds). It means that Albert still wants to receive PUSH messages.
4) Albert checks if any information about fast-food in the PUSH message list. And, he finds the advertisement of *Kruger Burger* and clicks its URL. Then, PTTP request (⑦) is delivered to *Kruger Burger*'s web server.
5) The web server sends PTTP responses (⑧) including web pages to Albert's client. Now, Albert can see the *Kruger Burger*'s web site in the PWW web browser.
6) Albert checks out the rough map of *Kruger Burger* in the web site. And he heads to the 3rd exit of subway station to stop by *Kruger Burger*.
7) As he arrives at *Kruger Burger*, he closes the client program. Before closing, the client program sends CLOSE message (⑨) is to the PUSH server. After receiving the message, the server deletes the connection information of Albert from its per-connection state list.

*2) Subway Arrival Time Notification Service (Fig. 8):* In scenario 2, Betty is planning to get on the subway for business trip. Unfortunately, she skipped lunch due to the busy schedule. At the moment, she discoveries the take out waffle house near the subway station. But, she is wondering if she can buy waffle without missing the subway. To know the subway

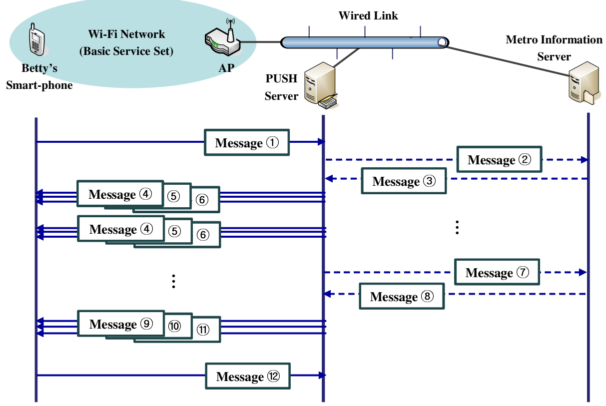


Fig. 8. PWW Scenario 1: Subway Arrival Time Notification Service

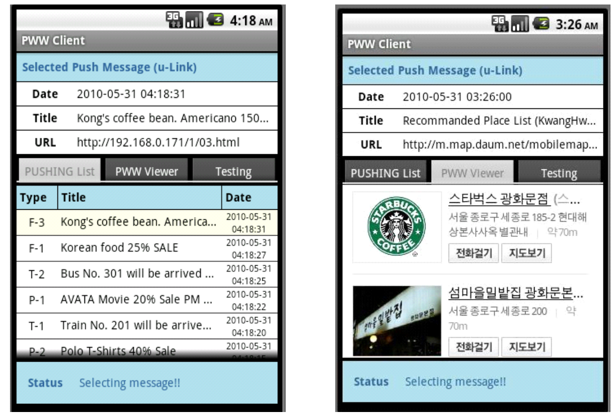


Fig. 9. Our Prototype of PWW Client (on Android 2.1 Platform)

arrival time, she executes PWW client program on her smart-phone. Then, the PWW system behaves follows:

1) Same as described in 1) of scenario 1 (①).
2) The PUSH server periodically obtains the information of subway arrival times around the station from the metro information server. (message ②, ③) This process can be implemented by existing technologies such as SOAP [17].
3) The PUSH server periodically pushes the PUSH messages (④, ⑤, ⑥) informing the Betty of the remaining time until next arrival of the subway.
4) Betty knows that subway No. 105 arrives after 6 minutes by checking one of the received PUSH messages (⑤).
5) She decides to buy waffle because the waffle is prepared in just 3 minutes.
6) At that time, the PUSH server obtains new information from the metro information server (message ⑦, ⑧). After that, PUSH server does same as described in 3) (message ⑨, ⑩, ⑪).
7) After buying waffle, she arrives at the platform and checks the PUSH message (⑩) saying subway No. 105 arrives after 2 minutes.
8) When Betty ends her client on her smart-phone, PWW service also ends in a same manner described in 7) of scenario 1 (⑫).

Although we explained above service scenarios separately, in fact, different services can be simultaneously provided by using single PUSH server. In PWW, PUSH server is an agent doing PUSHING for multiple stakeholders.

## III. Implementation and Experimental Results

We implemented the location specific advertisement service (scenario 1 in section II-B) in a simple prototype form. Our system consists of a PUSH server, PTTP web server, and PWW client on smart-phone device. This configuration is basically same as scenario 1 in section II-B. However, we made all different PTTP web sites to be operated in a single machine for a convenience. Fig. 9 shows our implementations of PWW client on smart-phone device. The hardware and software specifications of our system are listed in Table. I.

TABLE I
HW and SW Specification of Experimental Test Bed

| Component | Specification | |
|---|---|---|
| | Name | Description |
| PUSH Server | CPU | Intel Pentium 3.0GHz |
| | Memory | 1024MB RAM |
| | OS | Windows XP Service Pack 2 |
| PTTP Server | CPU | Intel Centrino 1.5GHz |
| | Memory | 2048MB RAM |
| | OS | Windows Vista |
| | Others | Apache Web Server 1.0 |
| PWW Client | Product | SCI-Phone N21 (Smart-phone) |
| | Platform | Android 2.1 |
| Access Point | Model | NETGEAR WG602 v03 |
| | PHY | IEEE 802.11g (54Mbps) |

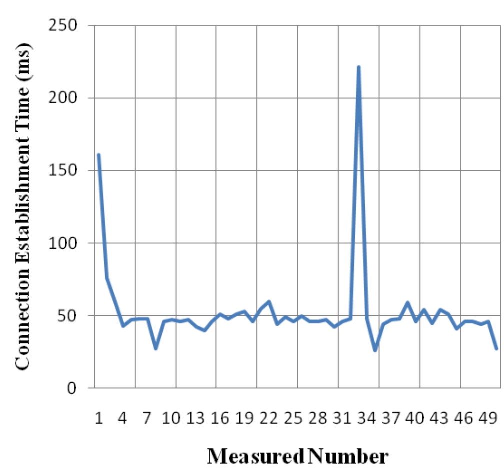


Fig. 10. Experimental Results on Connection Establishment Time

We performed experiments about the connection establishment time and PUSH message reception time of our PWW system. Fig 10 shows the amount of connection establishment time which is required for the process described in 1) of scenario 1 in section II. In Fig. 10, results are mainly distributed around 50ms. Two spike points are just due to the unexpected loss and retransmission of TCP segment. In fact, this phenomenon is commonly found in many applications using TCP/IP socket. Furthermore, the longest establishment time is less than 250ms. It is obvious that the connection establishment process is fast enough to satisfy the PWW user.

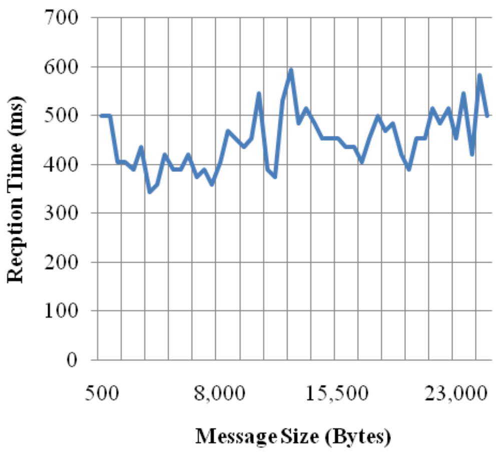


Fig. 11. Experimental Results on PUSH Message Delivery Time

Fig. 11 shows the PUSH message delivery time according to the message size from 500 bytes to 25000 bytes. As depicted in this figure, most of results are distributed around about 500ms although the PUSH message size is increasing. This is due to the pipelined feature of the TCP transmission algorithm. In TCP, the sender generally injects multiple packets into the network at once, and also receives multiple acknowledgements at once. So, message delivery time slowly increases as the size of data content increases. This implies PUSH sub-protocol can carry a relatively large amount of data in reasonable performance.

## IV. Conclusion and Future Works

Our Person Wide Web is a location based web service architecture which tries to maximize the LBS supportability by PUSHING, active notification about *geographically effective* web resources. Thus, PWW can be understood as a subset of the WWW which consists of location specific web resources based on the user's geographical position. In PWW, each marketer makes their own PUSHING document including link to their web site, then upload it to the PUSH server. And, local network service provider or data hosting provider take exclusively charge of the PUSH server management. Such cooperative service structure can create various business models where LBS provider, network operator and solution provider participate in. Thus, we believe that the PWW has a potential to create a new type of wireless data service market. We still have a long way to go before we can claim a success of PWW. First of all, we need to analyze common requirements of various wireless data service scenarios so that we construct robust specification of protocol to address them. In addition, it is also necessary to evaluate profitability of possible business models related with PWW. Finally, the key to success is commercialization. In the commercialization phase, additional research efforts about user satisfaction (ex. prevention of Spam PUSH) will be done. Currently, we are working for standardization of PWW. Further, we are planning to present extended results of our work with commercialized version of PWW system in the near future.

## Acknowledgement

This research is supported by the Ubiquitous Computing and Network (UCN) project, Knowledge and Economy Frontier R&D Program of the Ministry of Knowledge Economy (MKE) in Korea as a result of UCN's subproject 10C1-T1-20S. And, we are thankful to Sang-Ok Cho, research staff at Octacomm Inc. (http://octacomm.net) for his support to our research.

## References

[1] *IEEE Standard for Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, Nov. 1997, P802.11
[2] D. Astely, E. Dahlman, A. Furuska, Y. Jading, M. Lindstorm, S. Parkvall, "LTE: the evolution of mobile broadband", *IEEE Communications*, vol. 47, no. 4, April. 2009.
[3] *IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems*, Oct, 2004, P.802.16.
[4] E. Dahlman, S. Parkvall, J. Skold, P. Beming - 3G evolution: HSPA and LTE for mobile broadband, 2nd ed., Academic Press, 2008.
[5] http://www.wapforum.org/what/technical.htm
[6] http://en.wikipedia.org/wiki/Mobile_Web
[7] http://earth.google.com
[8] http://en.wikipedia.org/wiki/Navigation
[9] J. Raper, G. Gartner, H. Karimi, and C. Rizos, "A critical evaluation of location based services and their potential", *Journal of Location Based Services*, vol. 1, no. 1, Mar, 2007, pp. 5-45.
[10] J. Kjeldskov, C. Christiensen and K. Rasmussen, "GeoHealth: a location-based service for home healthcare workers", *Journal of Location Based Services*, vol. 4, no. 1, Mar, 2010. pp. 3-27.
[11] J. Delfos, T. Tan and B. Veenendaal, "Design of a Web-Based LBS Framework Addressing Usability, Cost, and Implementation Constraints", *Springer Journal of World Wide Web*, accepted for future publication.
[12] *HTTP/1.1 Speicifications, Internet Draft*, Mar, 2010.
[13] http://www.w3.org/TR/REC-html40/intro/intro.html
[14] P. Mockapetris, "Domain Names - Concepts and Facilities", *Internet Draft (RFC 2627)*, Nov., 1987.
[15] J. Kurose and K. Ross - Computer Networking - A top down Approach, 3rd Edl, Addison-Wesley, 2005.
[16] T. Dierks, E. Rescorla, "The Transport Layer Security (TLS) Protocol Versioin 1.2" *Internet Draft (RFC 5246)*, Aug, 2008
[17] http://www.w3.org/TR/soap