Seamless Application Push with Secure Connection

- System to realize effective usage of smart devices for business class sustainability -

Yosuke Nakamura†, Kazuaki Nimura† , Hidenobu Ito†, and Nobutsugu Fujino†

†Fujitsu Laboratories Ltd.
4-1-1, Kamikodanaka, Nakahara-ku, Kawasaki, Japan
{nkmr, kazuaki.nimura, itou.hidenobu, fujino}@jp.fujitsu.com

Abstract –We need to do jobs equally both inside and outside the office when using Application push. Here, push notifications should be reached on smart devices even if they are on different network domains. In addition, because push gateways that issue notifications deal with privacy and sensitive data and the application server stores applications including confidential data, these data should not be placed on the Internet. Therefore, when smart devices download applications from the Internet, a secure connection to a company's intranet is required. Notifications also need to be received while using secure connections.

We propose an architecture of seamless application push that enables notifications to be sent to smart devices regardless of whether they are on the Internet or intranet. We implemented the architecture using an Android smartphone and server. We confirmed from our evaluation that it could accomplish seamless push notifications and the required time through the intranet on the longest path was reasonably low at 157.4 [msec].

Keywords: Android, smart device, VPN, intranet, seamless push

1 INTRODUCTION

Both consumers and companies have been replacing their devices with smart devices [1] such as smartphones and tablets with today's rapidly expanding technology. However, users occasionally waste too much time in setting up such devices when they use them. We propose an "Application Push & Play (APnP)" to solve this issue, which is a concept of dynamic installing and executing applications without user operations for smart devices in IWIN2011 [2]. We also expanded this concept securely [3]. Push notifications represent the baseline technology when smart devices connect to push gateways (P-GWs) with transmission control protocol (TCP) sessions and smart devices can receive push notifications during continuous sessions [4]. The uniform resource locators (URLs) of allocated applications are transmitted using the notifications of push messages. After a push message is received, a smart device downloads an application indicated by a specified URL.

The system works well in situations where both smart devices and servers consisting of the P-GW to send push notifications and the application server to store applications are on the same network domain.

We need to do jobs equally both inside and outside the office in business, where push notifications should be reached on smart devices even if they are on different network domains. In addition, since P-GWs deal with privacy and sensitive data such as the identifiers and passwords of servicers, these data should not be placed on the Internet. i.e., P-GWs on the Internet should not contain private data.

Further, the application server should not be on the Internet either because business applications contain confidential data. Therefore, when smart devices download applications from the Internet, a secure connection to companies using services such as virtual private networks (VPNs) [5] would be required to connect to intranets. Push notifications also need to be received even when using VPNs.

2 RELATED WORK

Android cloud to device messaging (C2DM) [6] is a service that sends data from servers to their applications on Android devices. However, it does not deal with business uses. In addition, a Google account is required to use C2DM. This may not occasionally be preferable because some companies do not allow dedicated accounts to be used for business.

Mobile IP [7] is a network protocol that can transfer packets with the same IP addresses even if the destination of the device has moved to another network domain. There are two types of protocols: mobile IPv4 [8] and mobile IPv6 [9]. The home agent in the definition of mobile IPs is located on the same network domain as the sender and the foreign agent is on the same network domain as the receiver. A packet is capsuled between the home agent and foreign agent and is able to be sent to the receiver even if it is located at a different network domain. However, research has identified two issues with current mobile IPs that could be obstacles to their wider use [10]. The first is in providing a home agent where the network environment around it is faced with various difficulties, one of which is the home agent needs to have a global and fixed IP address of all smart devices, and the second is the conflict with current Internet security mechanisms such as firewalls [11]. In addition, another reason mobile IPs are not suitable for introduction is because they are implemented in network layer 3, which requires dedicated network switches.

3 PROPOSED ARCHITECTURE

This section summarizes the requirements described in Section 1 and presents the proposed architecture that meets these requirements.

The proposed architecture is the same as that for mobile IPs in terms of the necessity for servers both on the intranet and Internet. However, the proposed architecture has three main advantages that enable companies to adopt it more easily than mobile IPs for their network environment.

- It is unnecessary to replace network equipment. The proposed architecture is achieved in the application layer, and does not require the IP packet format to be changed. However, mobile IPs need the IP packet format to be extended because they use the IPv6 destination option. Therefore, network equipment such as routers and gateways must be renewed to use mobile IPs.
- No additional network ports are required for company firewalls. The proposed architecture uses HTTP for communication between the intranet and Internet. HTTP is typically permitted to pass communications on company firewalls. However, mobile IPs must use port number 434 in UDP for communications such as registration requests for terminals between the intranet and Internet. Nevertheless, this port number may not be permitted on company firewalls.
- Our architecture does not require to have a global nor fixed IP address of smart device because it is achieved in the application layer and assigns a destination address which is independent with the precious IP address.

3.1 Requirements

The three main requirements to implement the system are:

(a) Smart devices should receive push notifications regardless of whether they are on the Internet or intranet.

(b) They should be able to receive notifications even VPN connections are established.

(c) Neither confidential nor private data should be stored on the Internet.

3.2 Architecture

We propose an architecture for seamless application push that enables push notifications to be sent to smart devices regardless of whether they are on the Internet or intranet.

The architecture for the system is outlined in Fig. 1.

Four components are defined in APnP [2]. However, their use in this system is slightly different from that in conventional use.

- A private P-GW is equivalent to a conventional P-GW. A private P-GW is located on the intranet and accepts connections from the push client, receives requests for push notifications, and sends push notifications to smart devices. Private data such as user information and push messages are stored in the private P-GW.
- An application server stores applications and is located on the intranet.
- A push client is a function in a smart device that connects to the P-GW through a TCP session, receives push notifications, and transmits the URLs of applications through notifications. It extracts URLs from notifications if the notifications request download applications and the extracted URLs are passed to the application downloader.
- An application downloader is a function in a smart device and it downloads applications from the application server based on URLs.

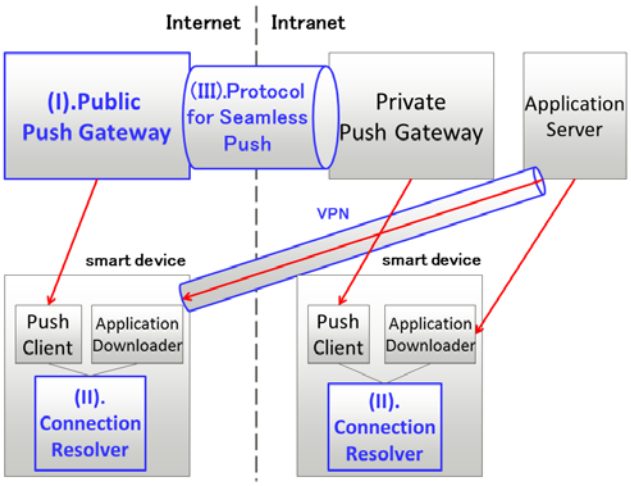


Figure 1: Architecture for Seamless Application Push

The proposed architecture has three additional new features:

(I) There is a public push gateway on the Internet in addition to a private P-GW that manages the connection to a smart device and notifies of push messages. The public P-GW is a contracted version of a private P-GW that particularly eliminates data related to privacy.

(II) A connection resolver identifies whether the network for a smart device is connected to the Internet or intranet and it switches the connection to a public P-GW or private P-GW. In addition, the connection resolver establishes a VPN connection to the intranet if necessary. When a smart device is on the intranet, it determines whether the push client should connect to the private P-GW without establishing a VPN connection to download an application. However, when a smart device is on the Internet, it determines that the push client should connect to the public P-GW and it makes the VPN client establish a VPN connection to download an application.

(III) A protocol for seamless push notification is used to synchronize the state of the private P-GW and public P-GW to maintain consistency.

The system should have features (I), (II), and (III) to satisfy requirement (a). A smart device connects to a public P-GW when it is located on the Internet and a private P-GW when it is located on the intranet (I). A connection resolver

detects where the smart device is and identifies which P-GW it should connect to (II). If the smart device is located on the Internet, the private P-GW receives a push request and forwards it to the public P-GW (III).

The system should have features (I) and (II) to satisfy requirement (b). When a smart device is located on the Internet, the connection resolver is responsible for establishing a VPN connection. Once a VPN connection is established, the connection resolver asks the push client to switch the connection to a private P-GW. After the VPN connection has terminated, the connection resolver asks the push client to switch the connection back to the public P-GW.

The system should have features (I) and (III) to satisfy requirement (c). Although private data are required to send push requests, the public P-GW should not store them or have an interface to register them. Therefore, the private P-GW receives the request and forwards it to the public P-GW by using the protocol for seamless push notifications. In addition, although the private P-GW has the delivery status for push notifications, the public P-GW should not have this because the delivery status includes privacy information. Consequently, the public P-GW needs to ask the private P-GW about its status.

4 IMPLEMENTATION

This section describes the components, protocols, and workflows for the implementation of the proposed system.

4.1 Components

Figure 2 outlines the implementation of the proposed system.

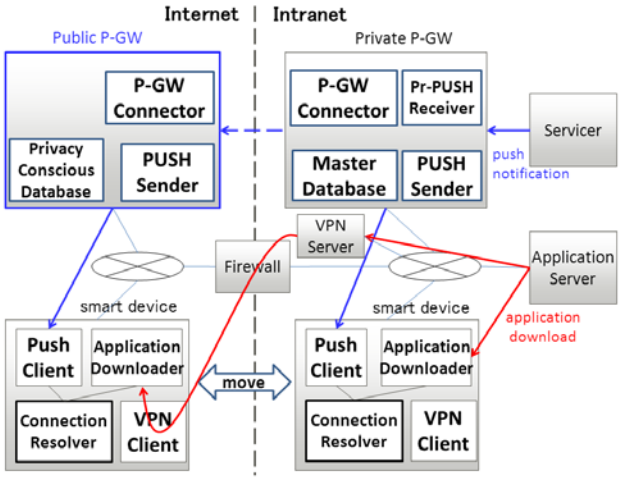


Figure 2: Implementation of proposed system

The system consists of five components: the private P-GW, public P-GW, application server, VPN server, and smart device. The private P-GW, the application server, and the VPN server are located on the intranet. The public P-GW is located on the Internet. The smart device moves between the Internet and intranet.

The private P-GW consists of four components.

The private push receiver (pri-push receiver) receives a push request and an inquiry as to whether a smart device is connected to the P-GWs or not from a servicer. The pri-push receiver authenticates the servicer in a push request and checks whether the destination address is pre-registered. The pri-push receiver checks whether the smart device is connected in an inquiry.

The push sender is an access point for the push client when the smart device is on the intranet. While the push client is connected to the push sender, the smart device can receive push notifications. It authenticates the smart device to prevent connection to malicicious devices. It also encrypts communication with the secure socket layer (SSL) to protect push notifications [12].

The P-GW connector handles the protocol for seamless push notifications. See the protocol for seamless push notifications in Subsection 4.2 and the workflow on the Internet in Subsection 4.3 for the protocol and its use.

The master database (MDB) manages private data such as those for the registered identifier and password for the servicer, the destination address that the private P-GW assigned to the smart device, and message queue including confidential data such as the URL of the in-house application required to send push notifications. These data should only be stored in the private P-GW. The pri-push receiver, push sender, and P-GW controller handle notifications through the MDB.

The public P-GW consists of three components.

The push sender is an access point for the push client when the smart device is on the Internet. It has the same functions as the private P-GW.

The privacy conscious database (PC-DB) is a DB that only manages the connection information of smart devices to send push notifications. There is no need to treat connection information as privacy information if the P-GW creates the connection information and changes it regularly because malicious users cannot relate the user to the smart device by using the connection information. The P-GW in our architecture creates the connection information and changes it regularly. Therefore, we do not need to treat the connection information as privacy information. There is no data shared between the MDB and PC-DB, so those DBs do not need to be synchronized for keeping the consistency.

The P-GW connector has the same functions as those for the private P-GW.

The smart device has four components for handling push notifications.

The VPN client manages to establish and disconnect a VPN connection by using the direction from the connection resolver. We need to be able to receive the direction from other applications to directly establish and disconnect VPN connections from the connection resolver.

The connection resolver contains the registered service set identifications (SSIDs) of access points on the intranet in advance. When the smart device connects to the network, the connection resolver checks the type of connection. For example, if the connection is third generation (3G) or long-term evolution (LTE), the connection resolver identifies the

Table 1: Protocol for seamless push

Protocol	Description	Source	Destination
FORWARD	Forward push notification from private P-GW to public P-GW. Public P-GW only accepts FORWARD command from private P-GW.	Private P-GW	Public P-GW
FORWARD _SUCCESS	If push notification to smart device is completed, public P-GW replies to private P-GW with FORWARD_SUCCESS.	Public P-GW	Private P-GW
REQUEST_ RESEND	Inquiry to private P-GW is issued to check whether or not push notification is resent to smart device, which is needed to connect to public P-GW stored in private P-GW to resend notification when smart device is connected to public P-GW.		

smart device is on the Internet. If the connection is Wi-Fi, it checks the SSID of the access point (AP) and confirms whether the SSID was registered, and then the connection resolver identifies the smart device is on the intranet. Otherwise, it considers the smart device is on the Internet. It requests the VPN client to establish a VPN connection for the intranet. If we can use VPN even when the smart device is on the intranet, the mode changes, and it switches the behavior of the client regarding whether the client is using the VPN or not. Therefore, it may not need our architecture. However, our architecture needs the mode to change because of the lower power consumption of the smart device. As we want to use the smart device as long as possible, lower power consumption is important. If we use VPN even when the smart device is on the intranet, the VPN client is constantly running and establishing a VPN connection. Therefore, unnecessary CPU and memory resources are wasted and the power consumption of the smart device increases. Consequently, we implemented changes to the mode because we minimized the opportunity of establishing VPN connections and decreased the power consumed by the smart device.

See Subsection 3.2 for the push client and application downloader.

4.2 Protocol for Seamless Push

Table 1 summarizes the protocols for seamless push conveyed by the hypertext transfer protocol (HTTP) POST. Because a firewall generally blocks most communications such as the transmission control protocol (TCP) of customer-defined sessions, we use the HTTP protocol that is typically permitted by the firewall.

4.3 Core Workflow for Proposed System

Our system has two P-GWs, a private P-GW on the intranet and a public P-GW on the Internet as seen in Fig. 2. However, the application server is only on the intranet. Moreover, the smart device moves between the intranet and Internet. Therefore, the smart device needs to identify whether it is on the intranet or Internet and it connects to a suitable P-GW to receive its push messages regardless of its position. The smart device also needs to establish a VPN connection if it is on the Internet to download applications regardless of its position. The choice of connectable P-GW and a necessary VPN connection are core workflows for our architecture. Therefore, we will explain these workflows.

First, we will explain the method of choosing connectable P-GW. The connection resolver needs to identify whether the smart device is correctly connected to the intranet or Internet to choose a suitable P-GW. First, the connection resolver checks information for the type of network. If the type of network is a mobile network such as 3G or LTE, the connection resolver identifies whether the smart device is on the Internet and the smart device connects to the public P-GW. If the type of network is a Wi-Fi connection, the identification is complex and the connection resolver needs to check the SSID of the Wi-Fi AP. The user sets the SSID preliminarily treated as the intranet to the connection resolver. If the SSID of the Wi-Fi AP fits the setup SSID to the connection resolver, the connection resolver identifies that the smart device is on the intranet and the smart device connects to the private P-GW. If not, the connection resolver identifies that the Wi-Fi connection is the Internet such as a hot spot and the smart device connects to the public P-GW.

For example, consider the case of setting “FUJITSU LAB” as the SSID to the connection resolver. If the smart device connects to the Wi-Fi network where the SSID is “FUJITSU LAB”, the smart device connects to the private P-GW. Other than this, the smart device connects to the public P-GW.

Next, we will explain the method of choosing whether the smart device has established a VPN connection or not when the smart device downloads applications.

The connection resolver also uses information on the type of network and the SSID of the Wi-Fi access point to identify whether it needs to establish a VPN connection or not. If the smart device connects to a mobile network or a Wi-Fi connection where the SSID is not the setup SSID to the connection resolver, the connection resolver identifies that the smart device is on the Internet and the smart device establishes a VPN connection when the smart device downloads applications. However, if the smart device establishes a Wi-Fi connection where the SSID is the setup SSID to the connection resolver, the connection resolver identifies that the smart device is on the intranet and the smart device does not establish a VPN connection. When the smart device establishes a VPN connection, it temporarily connects to the intranet. Therefore, the connection resolver makes the smart device connect to the private P-GW while it is establishing a VPN connection. In this way, the smart device can continuously receive push messages even when it has established a VPN connection.

For example, consider the case of setting “FUJITSU LAB” as the SSID to the connection resolver. If the smart device connects to a Wi-Fi network whose SSID is “FUJITSU LAB”, it does not establish a VPN connection and directly downloads applications from the application server. However, if the smart device connects to the mobile network or Wi-Fi network where the SSID is not “FUJITSU LAB”, it establishes a VPN connection and downloads applications. Further, the smart device connects to the private P-GW while it is establishing the VPN connection.

Lastly, we discuss the impact of SSID impersonation. When a Wi-Fi AP which has the same SSID defined in this system is placed, the smart device may connect to the wrong Wi-Fi AP. In this case, the push client cannot reach the P-GW and the smart device cannot receive a push notification. A way to avoid this issue is that the push client terminates the Wi-Fi connection if the push client fails to connect to P-GW in the Wi-Fi network and switches over to the mobile network for the continuous push service.

Our system receives push messages and downloads applications when the smart device is both on the intranet and Internet by using these methods. Furthermore, the smart device is establishing a VPN connection.

5 EVALUATION

This section presents the hardware, software, and network environments we used to evaluate the system and the performance of push notifications.

5.1 Qualitative Evaluation

First, we compared the proposed system with the conventional push system. The new system has two main advantages to the conventional push system. The first is its capabilities for seamless push as listed in Table 2.

Table 2: Capabilities for seamless push

	intranet, included via VPN	**Internet**
Proposed system	Yes: Refer to Subsection 4.3	Yes: Refer to Subsection 4.3
Conventional push system on intranet	Yes	No: *1
Conventional push system on the Internet	No: *2	Yes

*1: VPN connection is always required.
*2: Port used by conventional push is required to open in firewall.

“Yes” indicates the smart device can receive push notifications without a VPN connection and a firewall is established between the intranet and Internet when using push notifications. “No” means the smart device cannot receive push notifications.

The proposed system enables the smart device to receive push notifications regardless of whether it is connected to the Internet or an intranet. Even when it is connected to the intranet via VPN, the connection resolver switches to connect the private P-GW. Further, the smart device cannot receive notifications across networks by using the conventional push system.

Another advantage is protection of both confidential data and private data as summarized in Table 3.

Table 3: Protection of data

	intranet	**Internet**
Proposed system	Yes: see MDB in Subsection 4.1.	Yes: see privacy conscious DB in Subsection 4.1.
Conventional push system on intranet	Yes: *1	No: *2
Conventional push system on Internet	No: *2	No: *3

*1: DB stores private data, but DB is on intranet.
*2: Push notification cannot be received.
*3: DB stores confidential and private data.

“Yes” means that the system fulfills the first requirement that confidential data and private data are not stored on the Internet and the second that the smart device can receive push notifications both on the intranet and Internet.

“No” means that the system cannot fulfill either requirement.

The proposed system allows both confidential data and private data to only be stored on the intranet. No smart device can receive push notifications on the Internet without storing private data there while using the conventional push system as it is.

5.2 Quantitative Evaluation

We measured the delivery time for push notifications from an application server to a smart device that may affect user experience. Each component presented in Section 4 was operated on the following hardware to evaluate the system:

Private P-GW, Public P-GW, and Application Server:

- A Fujitsu LIFEBOOK E-8290 was used as the Private P-GW. It had an Intel Core2Duo processor T9600 operating at 2.80 GHz, 4 GB of main memory, and 160 GB of HDD.
- A Cent OS 6.2 [13] was used as the operating system (OS). Apache and the Java application server (Tomcat) were used for the Pri/Pub-push receiver and P-GW controller. The Pri/Pub-push receiver and P-GW controller were placed as servlets on top of the Java application server. A C-based program handled communication with the PC for the push sender. These were communicating with a socket between the three functions. MySQL was used for the DB. The application server was a general Web

server. Apache and the Java application server (Tomcat) were used.

Smart device:

- A Fujitsu F-10D [14] was used as the smart device. It had a 1.5-GHz quad core and had Android 4.0 installed that supported the VPN client application programming interface (API) [15]. Android native applications such as the connection resolver could control the establishment and disconnection of VPN connections by the VPN Client API.
- The push client, application downloader, connection resolver, and VPN client were created as Android native applications with Java.

VPN Server:

- A Fujitsu LIFEBOOK A550/A was used as the VPN server. It had an Intel Core i5 processor 540M operating at 2.53 GHz, 4 GB of main memory, and 128 GB of capacity on a solid state disk. Fedora was used as the OS.

Figure 3 outlines the network environment we used for the evaluation.

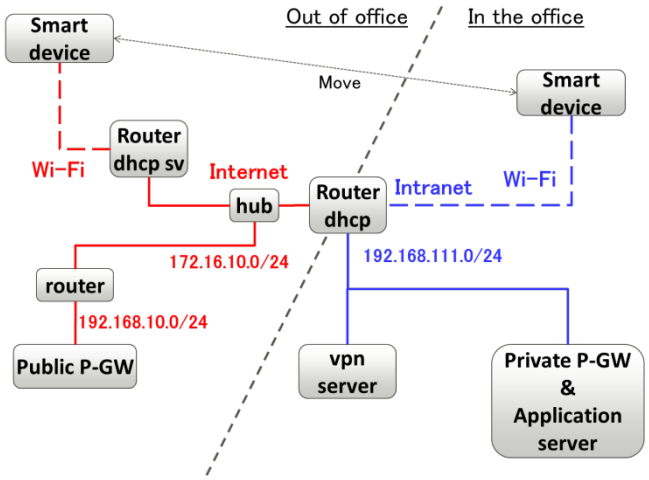


Figure 3: Network environment

We can see that it has two network segments. The first is treated as the intranet and the address is 192.168.111.0/24. The second is the Internet and the address is 192.168.10.0/24. Other than the HTTP for the protocol of seamless push application, it cannot transmit between the two networks. Refer to the hardware described above for the P-GWs, servers, and smart device. The information on network devices is as follows:

Router dhcp:

- A WN-G54/R3 was used as the router dhcp. It supported 100Base-TX/10Base-T as a LAN and 802.11b/g as a wireless LAN. It supplied the IP address of the intranet to the servers and the smart device as a Dynamic Host Configuration Protocol (DHCP) server and it assigned the IP address of the Internet from the Router dhcp sv. Only HTTP requests from the public P-GW to the private P-GW were accepted.

Router dhcp sv:

- The Aterm WM3400RN was used as the router dhcp sv. It supported 100Base-TX/10Base-T as a LAN and 802.11n/b/g as a wireless LAN. It supplied an IP address of the Internet to the router dhcp, the public P-GW, and the smart device.

Router:

- The Aterm LAN-W150N/RSPS was used as the router. It supported 100Base-TX/10Base-T as a LAN and 802.11n/b/g as a wireless LAN. It supplied an IP address to the public P-GW. HTTP requests from the private P-GW to the public P-GW and the TCP session from the smart device on the Internet were accepted. Although the public P-GW could directly connect to the Internet, a buffer such as a "demilitarized zone" (DMZ) is commonly organized for servers on the Internet. Therefore, we used a router.

Hub:

- FXG-05IMV was used as a hub that was used to increase the number of ports on the Internet.

First, we evaluated the system by using the delivery time for push notifications when the smart device was located on the Internet. Table 4 lists the results for the delivery times of push notifications both on the intranet and Internet when the servicer requested a thousand push notifications.

Table 4: Delivery times for push notifications

	Delivery time on intranet [msec]	**Delivery time on Internet [msec]**
Average	104.9	157.4
Maximum	391.0	387.0

The average time on the Internet was 157.4 [msec] and the maximum time was 387.0 [msec].

Application downloads came with the service by considering the use of APnP described in Subsection 6.1. An application download may take several seconds on the Internet using a 3G connection. However, the delay time is about 150 [msec] and it only has a small impact compared with the time for the download. Therefore, we concluded that the delay for the delivery time was acceptable.

Second, we compared the times notifications are received on the Internet and on an intranet to evaluate the performance of public P-GW. The average time is 104.9 [msec] on an intranet in Table 3 and the results for the Internet are inferior to this by 52.5 [msec]. An increase in the delivery time for push notifications was assumed because a public P-GW was added to the communication path on the Internet. We measured the processing time for the private P-GW, public P-GW, and communication when sending a push notification to verify whether the public P-GW caused the increase or not. Figure 4 and Table 5 have the results for processing times.

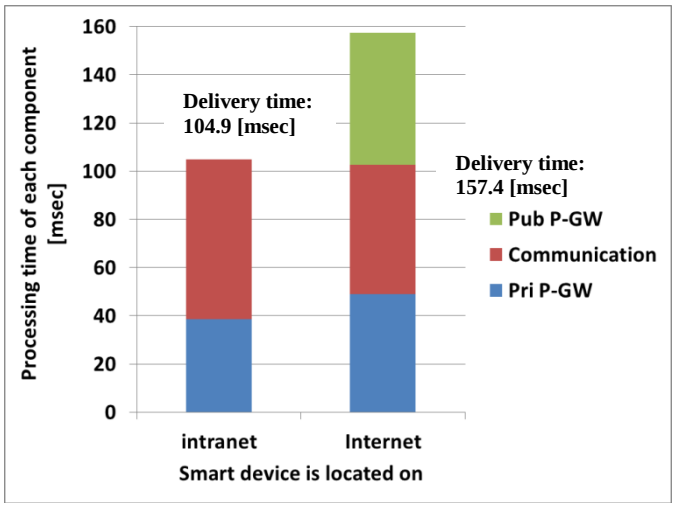


Figure 4: Processing times for each component

Table 5: Processing times

	Processing times on intranet [msec]	Processing times on Internet [msec]
Pri P-GW	38.6	49.1
Communication	66.3	53.4
Pub P-GW	-	54.9
Total	104.9	157.4

The results reveal that the total time for private P-GW and communication is almost the same for the intranet and Internet. Therefore, the times measured on the Internet for private P-GW and communication added to the processing time for public P-GW are longer than those for the intranet and are as expected.

The performance of push notifications was sufficient to function as APnP from these results and the structure of the proposed system and public P-GW was appropriate in terms of the delivery time for push notifications.

6 USE CASE

6.1 Use Case of Seamless Application Push

Figure 5 has an example of a use case for seamless application push, where the applications delivered with APnP, a URL of application is transferred by push notification then the smart device downloads the application from the URL transferred, are the editor of a presentation file on the intranet and the viewer of a presentation file on the intranet or Internet. Use only on the Internet is not assumed in this scenario. A user can only make a presentation file on the intranet using the editor and he or she can view a presentation file on the intranet or Internet. If the user views the presentation on the Internet, the file is downloaded via VPN from the server on the intranet.

The performance shown in Table 5 is enough that this use case should be workable. The requirement of this case is to maintain real-time notification to the smart device. A service requests a push notification when it detects the timing of sending an application. The delivery time is at most 150[msec] and the delay of push notification has no affect to keep the real-time notification.

Regarding the scalability, even if the number of smart device is increased, the public P-GW can accept the connection from many smart devices by increasing only the push sender. In addition, unlike mobile IP which has the issue on the home agent that has a bottle neck by the triangular routing and that would require sole DB [10], our architecture does not need to have the routing in the Public P-GW because it does not need to capture the change of IP address, so it can be achievable the scale by introducing additional independent DB.

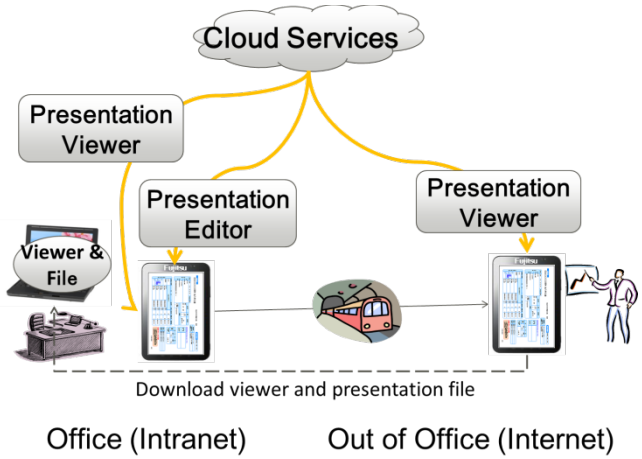


Figure 5: Use case of seamless application push

6.2 Multi-tenant Push Service

This section presents an extensive use case of the proposed system especially focusing on a general push service. Figure 6 outlines an example of multi-tenant push services that have multiple private P-GWs and a single public P-GW.

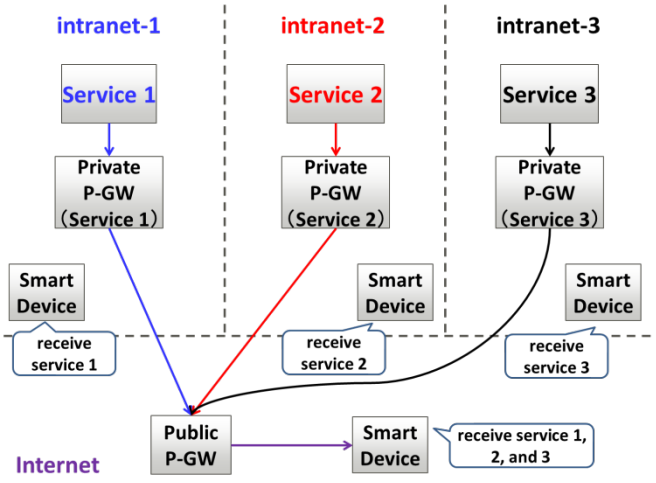


Figure 6: Example of multi-tenant push services

The public P-GW treats push notifications from all services to the smart device by the direction of each private P-GW. The smart device can receive all services on the

Internet and each service on each intranet. The smart device connects to the public P-GW and can receive all services on the Internet.

7 CONCLUSION

We proposed an architecture of seamless application push and implemented a system using an Android smartphone. We then confirmed that it could achieve seamless push notifications by which the smartphone could receive push notifications regardless of whether it was connected to the Internet or intranet and it could download applications from the application server. The smart device automatically could establish a VPN connection in conjunction with receiving a push notification. Usability was improved and the security of the smart device was maintained with our architecture because user interaction regarding the input of VPN passwords was unnecessary. In addition, we measured the delivery time for push notifications and the time we measured on the Internet was 157.4 [msec]. Our approximate requirements were within a second and the results were reasonably low. Future work would be to find ways to resume and continue downloading even a device has moved to a different network while downloading applications.

REFERENCES

[1] R. Cozza C. Milanesi, A. Zimmermann, T. Huy Nguyen, H.J. De La Vergne, S. Shen, A. Gupta, A. Sato, C.K. Lu, and D. Glenn, ``Market Share: Mobile Devices by Region and country, 4Q11 and 2011, Market Analysis and Statistics of Gartner,'' http://www.gartnet.com/DisplayDocument?ref=clientFriendlyUrl&id=1923316 (2012).

[2] H. Ito, K. Nimura, Y. Nakamura, A. Shiba, and N. Fujino, ``Application Push & Play -Proposal on Dynamic Execution Environment Combined with Personal Devices and Cloud Computing.-,'' Proc. of IWIN2011 (2011).

[3] K. Nimura, H. Ito, Y. Nakamura, and K. Yasaki, ``A Secure Use of Mobile Application with Cloud Service,'' Proc. of the 2nd International Workshop on Smart Mobile Applications in conjunction with PERVASIVE 2012 (2012). http://www.mobile.ifi.uni-muenchen.de/aktuelles/smartapps2012/smartapps12_secure.pdf

[4] K. Nimura, H. Ito, Y. Nakamura, A. Shiba, and N. Fujino, `` Proposal and Implementation of Pseudo Push Using Network Subsystem and Task Execution for PC,'' Proc. of IWIN2011 (2011).

[5] Virtual Private Network (VPN). http://en.wikipedia.org/wiki/Virtual_private_network

[6] Android Cloud to Device Messaging Framework. https://developers.google.com/android/c2dm/

[7] C. Perkins, ``IP Mobility Support,'' IETF RFC2002, http://www.ietf.org/rfc/rfc2002.txt (1996).

[8] C. Perkins, ``IP Mobility Support for IPv4, Revised,'' IETF RFC5944, http://www.ietf.org/rfc/rfc5944.txt, (2010).

[9] D. Johnson, C. Perkins, and J. Arkko, ``IP Mobility Support in IPv6,'' IETF RFC3775, http://www.ietf.org/rfc/rfc3775.txt (2004).

[10] M. Ishiyama, A. Inoue, T. Okamoto, and F. Teraoka, ``A study of the current Mobile IP status, the obstacles of its wide usage, and the future direction of mobility support'' (1998).

[11] F. Baker, and P. Savola, ``Ingress Filtering for Multihomed Networks,'' IETF RFC3704, http://www.ietf.org/rfc/rfc3704.txt (2004).

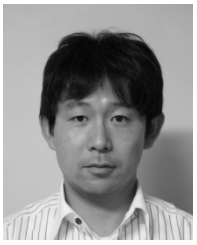
[12] T. Dierks, and C. Allen, ``The TLS Protocol Version 1.0,'' IETF RFC2246, http://www.ietf.org/rfc/rfc2246.txt (1999).

[13] The Community ENTerprise Operating System, http://www.centos.org/.

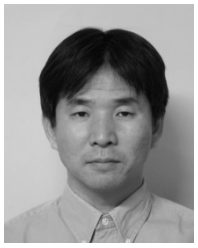
[14] F-10D, http://www.fmworld.net/product/phone/f-10d/spec.html?fmwfrom=f-10d_index

[15] Android 4.0 Platform Highlights, Android Developers, http://developer.android.com/sdk/android-4.0-highlights.html, 2011

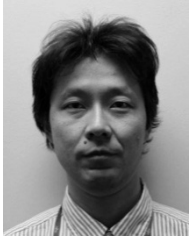
(Received October 18, 2012)
(Revised December 28, 2012)



Yosuke Nakamura received his BE and ME from the Graduate School of Engineering at Yokohama National University, Japan, in 2002. He joined Fujitsu Laboratories Ltd. in 2002. His current research interests include advanced technologies in personal computers and human centric computing.



Kazuaki Nimura received his BE and ME from the Graduate School of Information and Communication Engineering of Tokyo Denki University, Japan, in 1994. He joined Fujitsu Limited in 1994 and transferred to Fujitsu Laboratories Ltd. in 1997. His current research interests include advanced technologies for smart devices and human centric computing.



Hidenobu Ito received his BE and ME in Mathematical Sciences from the University of Osaka Prefecture, Japan, in 1993. He joined Fujitsu Laboratories Ltd in 1993. His current research includes mobile computing and human centric computing

Nobutsugu Fujino received his B.S. and M.E. in electronics engineering from the University of Osaka Prefecture in 1986. He also received his Ph.D. in informatics from Shizuoka University in 2008. He joined Fujitsu Laboratories Ltd. in 1986. Since then, he has been engaged in radio communication systems and mobile computing, and is currently a research manager in human centric computing and multi-device interaction technology. His research interests include mobile and ubiquitous computing and network applications. He received the IPSJ Industrial Achievement Award in 2003.