International Journal of Computer Networks and Applications (IJCNA)
Volume 2, Issue 6, November – December (2015)

RESEARCH ARTICLE

# Design and Implementation for SIP-based Push-to-Talk Services over 802.11 Networks

Wen-Shiung Chen
VIPCCL, Dept. of Electrical Engineering, National Chi Nan University, Nan-Tou, Taiwan

Lili Hsieh
Dept. of Information Management, Hsiuping University of Science and Technology, Taichung, Taiwan

Ying-Neng Hsieh
VIPCCL, Dept. of Electrical Engineering, National Chi Nan University, Nan-Tou, Taiwan

**Abstract – Push-to-Talk (PTT) is a service that allows users using mobile phones in a way like walkie-talkie. All big telecommunication vendors and operators consider PTT an emerging application. 3GPP and OMA regulate the PTT specifications on the basis of Session Initiation Protocol (SIP) to assure interoperability. However, current PTT services are dedicated to cellular phones. We would like to realize PTT services with popular 802.11 wireless networks and VoIP services. Thus, in this paper, we implement a PTT system that works in 802.11 networks. Moreover, we interconnect it with GPRS network to compare the performance in-between these two heterogeneous networks. Simulation result shows that the latency to hear the talk calling from GPRS network to 802.11g network is only about 1~2 seconds while that of from 802.11g network to GPRS network is 6 seconds.**

**Index Terms – Push-to-Talk (PTT), Session Initiation Protocol (SIP), Wi-Fi Network, 802.11.**

## 1. INTRODUCTION

As more and more telecommunication products and technologies [1][2] thrive in the market, and Internet becomes prevalent, people can communicate and exchange information more instantly than ever before. Walkie-talkie allows users to push to talk right away, that is still common in use nowadays. It works in a two-way but half-duplex manner. The Push-to-Talk (PTT) over cellular networks allows users treating cellular phones like walkie-talkie and communicating with others in one-to-one or one-to-many manners [3]. Major telecommunication vendors, e.g., Nokia, Ericsson, Siemens, Motorola, and so on, consider PTT a killer application and keep providing time-to-market solutions. However, they encounter interoperability issues. Thus, the world's leading telecommunication and information technology companies, and mobile operators formed the Open Mobile Alliance (OMA) in 2002, to regulate specifications in order to support the interoperable end-to-end mobile services. In terms of PTT service, those are Push to-talk over Cellular, PoC) [4-6] specifications. For some vendors, e.g., Nokia, PoC is labeled as PTT [7] as well. PTT specifications primarily adopt the Session Initiation Protocol (SIP) for signaling control. Moreover, the 3rd Generation Partnership Project (3GPP) also adopts OMA specifications to be its own PoC standards [8]. In this paper, we mainly follow OMA PoC specifications to implement the PTT service over 802.11 networks on the basis of SIP. Through the implementation, we explore the floor and presence state control, and assess the delay caused by signals and messages. At last, we use dual-mode handsets for field test in-between GPRS and 802.11 networks to check the interoperability and compare the performance in-between these two heterogeneous networks.

## 2. TECHNICAL BACKGROUND

### 2.1. Session Initiation Protocol

#### 2.1.1. Protocol Overview

Session Initiation Protocol (SIP) [9] is an application-layer protocol which is similar to Hyper Text Transfer Protocol (HTTP) [10]. SIP is mainly used for call control, i.e., to establish, modify, and terminate a multimedia session in-between two end points, as shown in Figure 1. Thanks to SIP messages are based on text, users can easily read the message contents. Furthermore, developers can implement relevant applications more flexibly. As a result of flexibility, SIP is the most dominant signaling protocol for VoIP. SIP provides the following major functions,

1. To get the IP address of a device,
2. To invite end devices to join media sessions,
3. To use Session Description Protocol (SDP) [11] to exchange media information (e.g., audio codec),
4. To modify parameters of on-going media sessions (e.g., audio codec),
5. To terminate the media sessions.

#### 2.1.2. SIP Messages and Servers

SIP servers handle User Agents' (UA) request and response messages. There are three typical types of SIP servers: proxy, redirect, and registrar [9] that are depicted as follows.

**RESEARCH ARTICLE**

1. A proxy server mainly transfers and routes incoming messages sent by user agents or other proxy servers, to next proxy server or user agents by looking up databases and resolving the real IP addresses of the URIs of request messages.
2. A registrar server authenticates UA registration data. Once authenticated, the server records the real IP addresses of registered user agent into its database for proxy servers to look up. In practice, they are usually co-located with proxy servers.
3. A redirect server is similar to the proxy server. However, instead of helping with forwarding the request message to next stops, the redirect server simply looks up the database and replies the new locations to the user agent that sent the request. The user agent must send its request to the new next stop over again on its own.

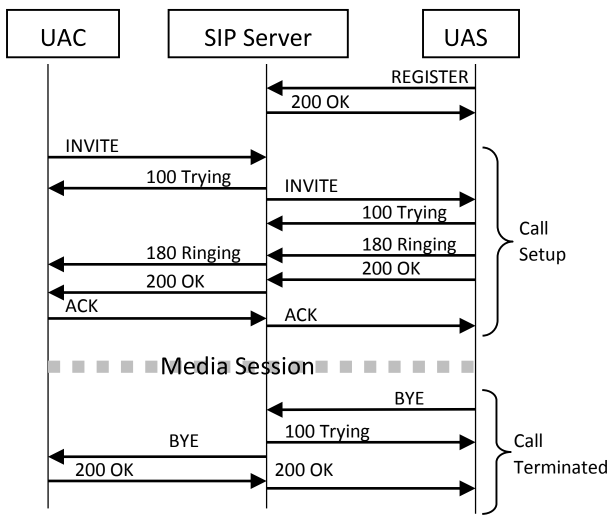


Figure 1 SIP messages and the life cycle of a call

There are two logical entities: the user agent client (UAC) sends request messages, and the user agent server (UAS) replies response messages. The request messages are sent by six main SIP methods [9], as follows.

1. The INVITE method is mainly used for establishing a media session in-between two end points.
2. The REGISTER method is used by user agent to register its own SIP URI and reveal its current IP address to the SIP registrar server.
3. The ACK method is used by the user agent to acknowledge its own INVITE transaction.
4. The CANCEL method allows user agent to cancel its invitation call before the recipient responds "200 OK" message to answers the call.
5. The BYE method is used for terminating an on-going media session.
6. The OPTIONS method allows user agent to query and determine the capabilities of another UA or proxy server, e.g., the supporting SIP methods, content types, and so on.

## 2.2. Push-to-Talk Architecture

Based on the PoC standards [4] of OMA and 3GPP, the overall system architecture can be categorized into the following major components: PTT server, PTT client, shared group/list/policy XML data management server (XDMS), presence server, SIP/IP core network, device management server, and aggregation proxy server, as shown in Figure 2.

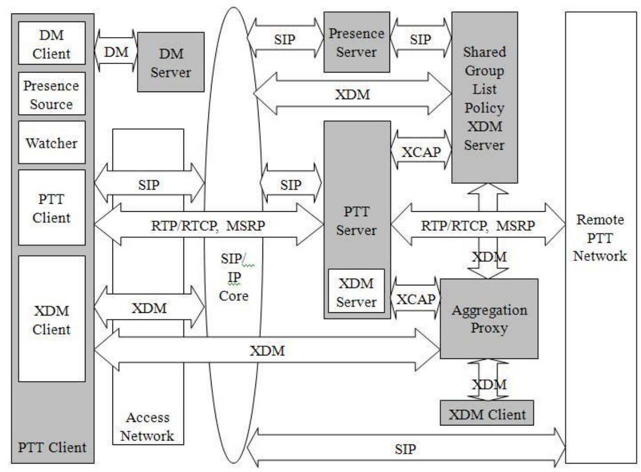


Figure 2 Push-to-Talk Architecture Overview

1. The PTT server mainly takes charge of session handling. Prior to help a client with invite online members to join its session, a PTT server must get the client's group member list, preferences (e.g., black and white lists, and answering mode) from the XDMS server, and present states of group members from presence server. Besides, the PTT server also arbitrates talk bursts, i.e., grants which group member to talk, and transfers media streams to the other invited PTT clients.
2. The PTT client is a user wireless device, such as a mobile handset, a PDA, and etc. Through SIP/IP core networks, the PTT client can access the XDMS server, PTT server, and presence server. Meanwhile, the PTT client can transmit and receive voice data to and from PTT servers via RTP/RTCP [12].
3. The shared group/list/policy XML data management server (XDMS) manages contact lists, group memberships, and access rights in XML formats.
4. The presence server can accept and store PTT user presence information, and forward such information to other users.
5. The SIP/IP core network is on the basis of IP, and uses SIP for signaling control, including proxy, registrar, and redirect servers as mentioned earlier. It provides user authentication and routes SIP messages correctly to PTT-related servers.

**RESEARCH ARTICLE**

6. The device management (DM) server is used for initializing and updating profiles of client devices.
7. The aggregation proxy server is to present PTT client a unified access point. It can forward clients requests to the right XDMS, and can execute extra functions, e.g., billing calculation, XML data compression and so on.

## 3. SYSTEM DESIGN AND ARCHITECTURE

### 3.1. Pre-established Sessions

The PTT client talk bursts are sent to the PTT server, and then the server distributes the bursts to other invited group members. Based on suggestions of PoC specifications, a PTT client must register to the SIP/IP core network before using the PTT service, as shown in Figure 3. Once registered, the client uses SIP PUBLISH method [13] to update its own presence state, as well as receives its buddy list and presence states of buddies. The user can check the presence information of buddy list to engage one-to-one or one-to-many group talking. Owing to the presence information and SIP signaling messages are all in all text-based formats, it may cause the time sensitive service considerable amount of delay.

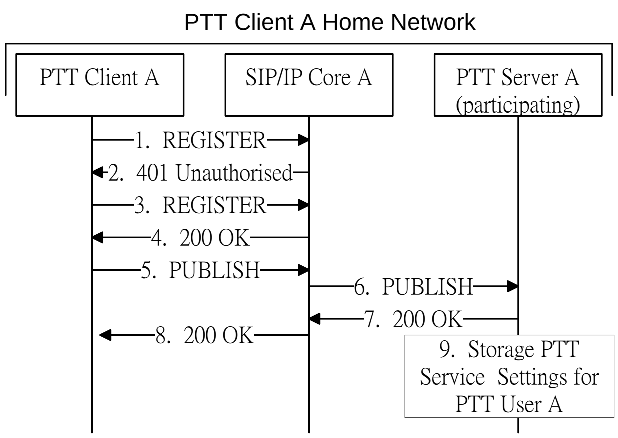


Figure 3 SIP registration of a PTT client

For instance, when a PTT user pushes the talk button and tries to talk with other group members, the size and number of delivered SIP messages during the call establishment are considerable. Particularly, in a wireless environment, it may consume considerable bandwidth for exchanging text-based messages, compared to binary messages. Besides, the user may start to speak as soon as pushes the talk button. However, due to the delay resulted from the exchanges of text-based messages; it may cause some initial part of voice data lost. Thus, according to the recommendation of OMA PoC, we utilize pre-established session to reduce the delay. After registration, PTT clients may send SIP INVITE messages to PTT servers ahead in order to complete the message exchanges that will take place for negotiating media parameters afterward, as shown in Figure 4.

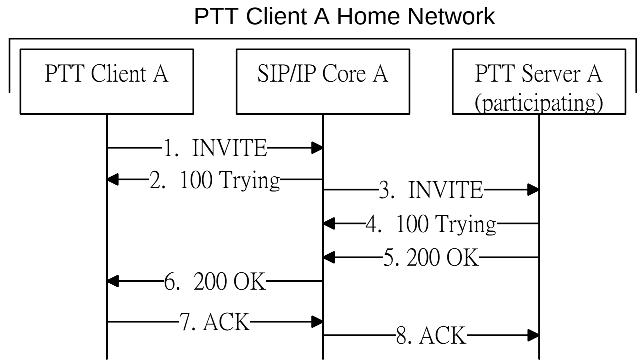


Figure 4 Pre-established session

### 3.2. Process of Talk Burst

Once both the originating (e.g., client A) and terminating (e.g., client B) parties finished the SIP registration process, the PTT client A can use SIP REFER method [14] to ask PTT server for inviting the other PTT client (e.g., the client B) to join its pre-established session, as shown in Figure 5. On the other hand, given that the client B have also already pre-established its session with a PTT server, if client B accepts the invitation, it will use SIP NOTIFY method [15] to inform client A as well as shown in Figure 6.

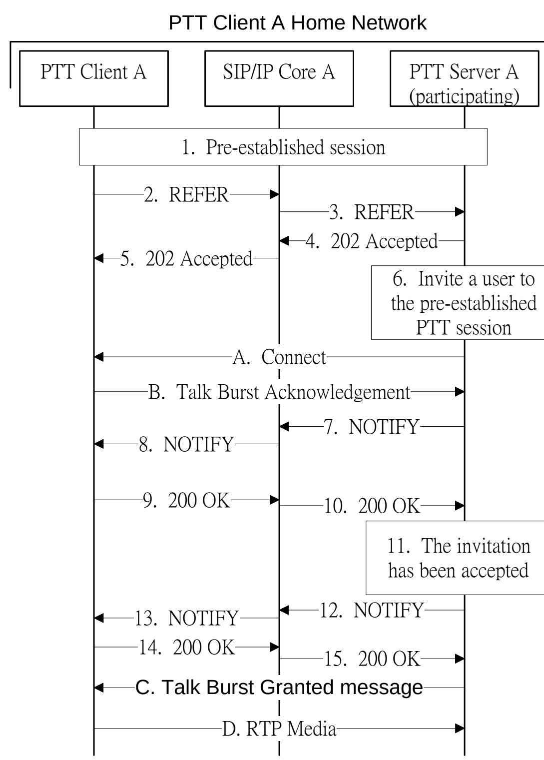


Figure 5 Talk burst at originating side

International Journal of Computer Networks and Applications (IJCNA)
Volume 2, Issue 6, November – December (2015)

**RESEARCH ARTICLE**

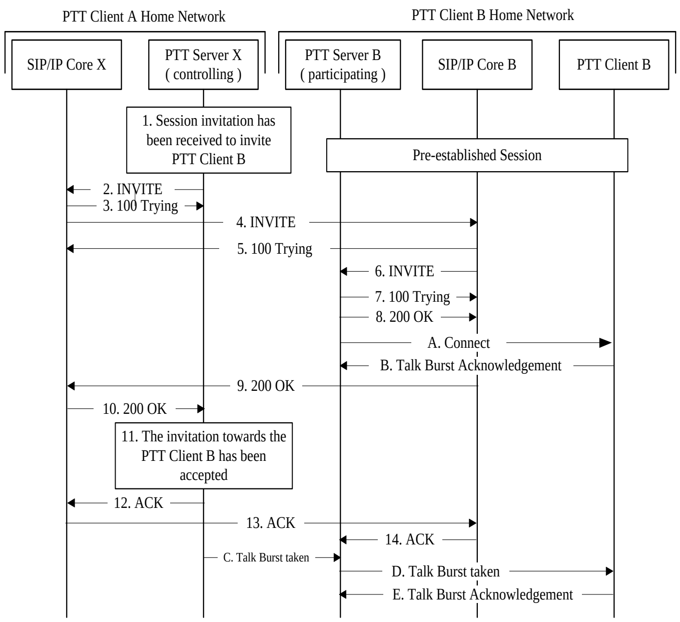


Figure 6 Talk burst at terminating side

## 3.3. Floor Control

During the talking, the PTT service is similar to walkie-talkie, i.e., a group member must push the talk button to ask the PTT server for permission to talk. The voice data is transferred via Real-time Transport Protocol (RTP) [12]. The talk burst control (also as known as floor control) messages are realized via the Talk Burst Control Protocol (TBCP) [6]. The TBCP messages are transferred via the APP format of RTP Control Protocol (RTCP) [12]. As shown in Figure 7, once granted, the talking PTT client can start to send the talk bursts. Meanwhile, the other group members can listen to the talk bursts. They cannot compete to talk until the talking PTT client releases the talk button as shown in Figure 8.

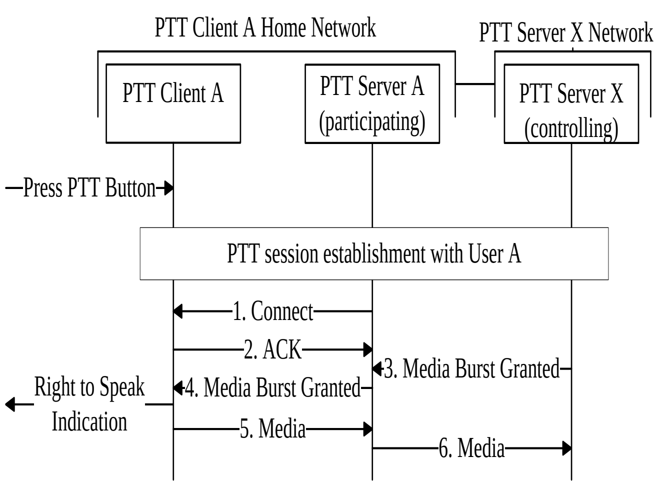


Figure 7 Talk burst control

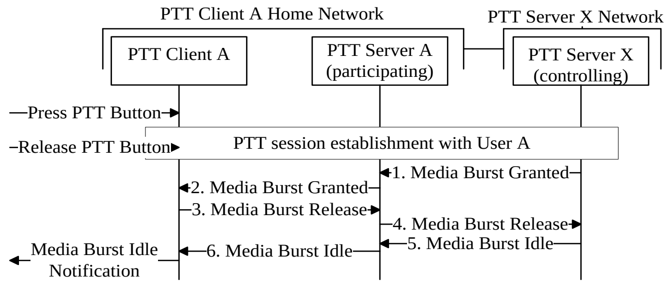


Figure 8 Release talk button

## 3.4. Presence State Control

During the PTT session, a PTT client can use SIP SUBSCRIBE method [15] and NOTIFY method [15] to acquire participant information, as show in Figure 9.

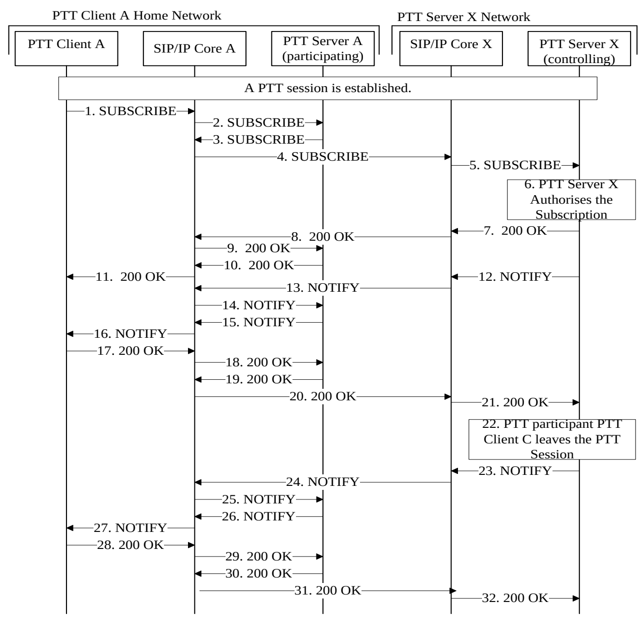


Figure 9 Subscription to participant information

## 3.5. Instant Message Delivery

A PTT client can use SIP MESSAGE method [16] to deliver personal alert instantly, as shown in figure 10. It can use SIP Message method to send group advertisement as well as shown in Figure 11.

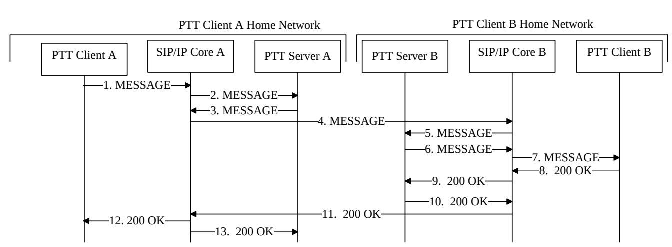


Figure 10 Instant personal alert

RESEARCH ARTICLE

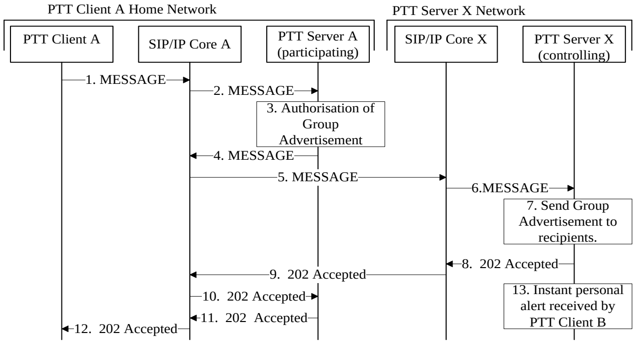


Figure 11 Group advertisement at originating side

## 4. IMPLEMENTATION AND FIELD TEST

### 4.1. Implementation Results

Like [17], we chose dual-mode handsets that support both GPRS and 802.11 as well as Java ME [18], to implement and practice the PTT client software. Both originating and terminating PTT clients stay in the 802.11 networks. The servers stay in the wired fast Ethernet local area network. We adapted openSER [19] for PTT server, XDMS, presence server and so on. After registration, the talking PTT client can choose and push to talk the participants, as shown in Figure 12.

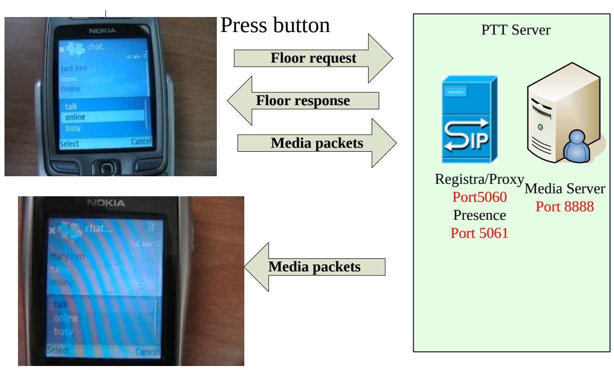


Figure 12 Snapshots

### 4.2. Performance Observation

During the test, we did not take firewall, network address translation (NAT) [20], or user mobility into consideration. Meanwhile, we followed the evaluation of delay analysis suggested by 3GPP [6], i.e., to analyze the latency from the moment to push button, to the moment that starts to send out the talking burst (i.e., the talking PTT client receives the Talk Burst Confirm message). Owing to the amount of the TBCP-related RTCP and session-related SIP messages are far less in proportion to the total amount of signaling messages; we summarize the most likely delay stages, as shown in Figure 13.

1. To request the radio resource of access network, supposed that the calling PTT client have been idle for a while.
2. To get the interior routing information of access network.
3. To send the SIP REFER message from the calling PTT client to the PTT server; and to send the SIP INVITE message from the PTT server to the called PTT client.
4. To automatically respond the answer from the PTT server that serves the called PTT client, in automatically answering mode.

We use packet capture software [22] to observe the latency of outcomes. Taking 802.11g for example, the total elapsed time of the first two items mentioned above, is less than 1 second, and so is the one of the rest two items.

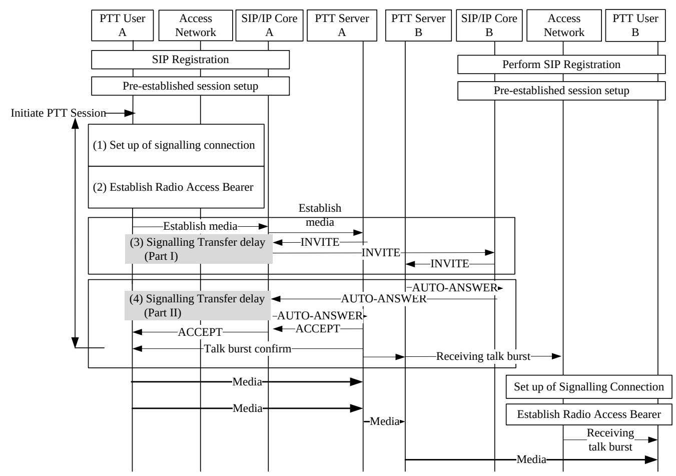


Figure 13 Potential delay analysis

### 4.3. Field Test with Heterogeneous Networks

To test the interoperability with different types of networks, we connect our client devices to GPRS networks, in addition to 802.11. The originating and terminating PTT clients are attached to GPRS and 802.11 networks respectively. The PTT server still stays in the wired campus networks. As shown in Figure 14, we go for field test to observe and record the actual outcomes. Each test lasts 3 seconds. The results are shown in the table 1. As shown in Table 1, it clearly indicates that the performance of GPRS impact the PTT service significantly.

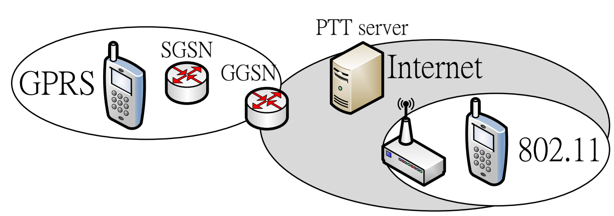


Figure 14 Interoperability in-between heterogeneous networks

RESEARCH ARTICLE

| Type | Latency to hear the talk |
|---|---|
| 802.11g(caller) → GPRS(callee) | ~ 6 seconds |
| 802.11g(callee) ← GPRS(caller) | About 1~2 seconds |

Table 1 Field test

## 5. CONCLUSION

In the 802.11 networks, we use SIP to implement PTT service, by adapting openSER for PTT server, and utilizing dual-mode handsets to run our client software. Besides, we do field test with GPRS network to check the interoperability in-between these two heterogeneous networks. The signaling performance over 802.11 is better than the one over GPRS. However, in practice, if the client device crosses different 802.11 access points, the hand-off issue will take place. Therefore, to solve the hand-off issue is indeed worth further discussion.

## REFERENCES

[1] R. Das and G. Tuna, "Machine-to-machine communications for smart homes," *International Journal of Computer Networks and Applications*, vol. 2, no. 4, pp. 196-202, 2015.
[2] H.-T. Chu, L. Hsieh and W.-S. Chen, "A novel design of instant massaging service extended from short message service with XMPP," *International Journal of Computer Networks and Applications*, vol. 2, no. 1, pp. 35-40, 2015.
[3] L. A. DaSilva *et al*., "The resurgence of push-to-talk technologies," *IEEE Communication Magazine*, vol. 44, no. 1, pp. 48-55, Jan. 2006.
[4] OMA Push to talk over Cellular (PoC) - Architecture, http://www.openmobilealliance.org/release_program/poc_v1_0.html.
[5] OMA Push to talk Over Cellular Control Plane Specification, http://www.openmobilealliance.org/release_program/poc_v1_0.html.
[6] OMA Push to talk Over Cellular User Plane, http://www.openmobilealliance.org/release_program/poc_v1_0.html.
[7] http://www.nokia.com/NOKIA_COM_1/About_Nokia/Press/White_Papers/pdf_files/whitepaper_pushtotalk_technology.pdf.
[8] 3GPP TR 23.979; 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3GPP enablers for Open Mobile Alliance (OMA); Push-to-talk over Cellular (PoC) services; Stage 2; http://www.3gpp.org/ftp/Specs/html-info/23979.htm.
[9] J. Rosenberg *et al*., "SIP: Session Initiation Protocol," IETF RFC 3261, 2002.
[10] R. T. Fielding *et al*., "Hypertext Transfer Protocol -- HTTP/1.1," Internet RFC 2616, Jun. 1999.
[11] M. Handley and V. Jacobson, "SDP: Session Description Protocol," IETF RFC 2327, Apr. 1998.
[12] H. Schulzrinne *et al*., "RTP: A Transport Protocol for Real-Time Applications," IETF 3550, Jul. 2003.
[13] A. Niemi, "Session Initiation Protocol (SIP) Extension for Event State Publication," IETF RFC 3903, 2004.
[14] R. J. Spark, "The Session Initiation Protocol (SIP) Refer Method," IETF 3515, April 2003.
[15] A. Roach, "Session Initiation Protocol (SIP)-Specific Event Notification," IETF 3265, Jun. 2002.
[16] B. Campbell and *et al*., "Session Initiation Protocol (SIP) Extension for Instant Messaging," IETF 3428, Dec. 2002.
[17] L.-Y. Wu *et al*., "A client-side design and implementation for push to talk over cellular service," *Wireless Communications and Mobile Computing*, vol. 7, no. 5, pp. 539-552, Jul. 2006,
[18] Java Platform, Micro Edition (Java ME), http://java.sun.com/javame/index.jsp.
[19] OpenSER - the Open Source SIP Server, http://www.openser.org/.
[20] W.-P. Chen *et al.*, "Performance measurement, evaluation and analysis of push-to-talk in 3G networks," *IEEE International Conference on Communications*, pp. 1893-1898, Jun. 2007.
[21] A. Balazs, "Push-to-talk performance over GPRS," *7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems, QoS in Wireless Networks*, pp. 182-187, Oct. 2004.
[22] Wireshark, http://www.wireshark.org/.