

An Implementation of IMS Based PoC Service Deployment

Jae-Oh Lee^{*}, Hong-Kyu Lee¹

¹School of Electrical, Electronics and Communication Engineering,
Korea University of Technology and Education

IMS 기반의 PoC 서비스 전개 구현

이재오^{*}, 이흥규¹

¹한국기술교육대학교 전기전자통신공학부

Abstract The IP Multimedia Subsystem(IMS) is a framework that provides access to the content of Internet and Telecom services anytime and anywhere with guaranteed Quality of Service(QoS) and manageability by separating control functions from bearer and services. The Service Delivery Platform(SDP) provides common interfaces and protocols to deploy existing or new services in an efficient way. Therefore SDP over IMS plays a role of bridge between established network and new IMS network by simplifying the interaction among application services. In order to enrich the multimedia network communication, we try to deploy the Push-to-talk over Cellular(PoC) service which is considered as the outstanding and distinguished half-duplex Voice over IP(VoIP) application service among deployable candidate services over mobile network. In this paper we investigate the advantages of PoC service and PoC architecture firstly, and then focus on the its practical implementation for the prototype to validate the feasibility of its deployment and realization.

요약 IMS(IP Multimedia Subsystem)는 전송 및 서비스들로부터 제어 기능들을 분리하도록 설계되었으며, 이러한 구조를 바탕으로 보장된 QoS(Quality of Service) 및 관리 기능을 수행하여 언제 어디서나 인터넷의 콘텐츠 및 텔레콤 서비스들을 접근할 수 있도록 제공하는 프레임워크이다. SDP(Service Delivery Platform)는 기존 및 신규 서비스들을 효율적인 방법으로 전개하기 위하여 공통의 프로토콜 및 인터페이스를 제공한다. 따라서 IMS 상의 SDP는 응용 서비스들 사이의 상호작용을 단순화하여 수행할 수 있으며, 이러한 인터페이스를 통하여 기 구축된 네트워크와 신규 IMS 네트워크 사이에서 교량 역할을 수행한다. 멀티미디어 네트워크 통신을 풍부하도록 구축하기 위하여, 모바일 네트워크상에서 전개될 수 있는 후보 서비스들 가운데 대표적이며 독특한 반 이중 방식의 VoIP(Voice Over IP) 응용 서비스로 고려되는 PoC(Push-to-talk over Cellular) 서비스를 전개하려 한다. 본 논문에서는 PoC 서비스 및 구조의 장점을 조사하고, PoC 서비스 전개를 통한 실행을 통한 가능성 및 실현을 확인하기 위하여 프로토타입(prototype)을 구축하는 실질적 구현에 집중하였다.

Keywords : IMS, PoC, Presence Service, SIP

1. Introduction

The IP Multimedia Subsystem(IMS) is an architectural framework for delivering Internet Protocol(IP) multimedia services to mobile users[3,8].

It includes functionality for end user authentication and authorization, call control and charging for multimedia sessions, as well as Quality of Service(QoS) decision and notifications at the data path level through the integration with core network platforms such as the

^{*}Corresponding Author : Jae-Oh Lee (Korea University of Technology and Education)

Tel: +82-2-560-1355 email: jolee@koreatech.ac.kr

Received April 8 2015

Revised (1st April 17, 2015, 2nd April 30, 2015)

Accepted July 16, 2015

Published July 31, 2015

3GPP Evolved Packet Core(EPC). In communication and network businesses, architectures and systems are independently with existing service or new service deployment without considering common functions and services between them. The Service Delivery Platform(SDP) provides common interfaces and protocols to enhance the IMS application service layer with extended services. What the SDP focuses on is service creation, deployment, delivery and management. So, the SDP over IMS plays such a role of bridge between established network and new IMS network by simplifying the interaction among the services.

The PoC is the outstanding half-duplex Voice over IP(VoIP) technology that enables traditional walkie-talkie service over mobile networks and one among services in the SDP over IMS. Instead of being conversational, PoC calls tend to be instructional and directional, with immediacy of connection without the delays of call setup and dialing. Building the PoC service over the exiting wireless networks gives the operator an opportunity to reduce the needed investments and allow forward compatibility to VoIP[5]. To end users, its ‘always-online’ solution offers an affordable and virtually instant connection to individuals or groups that allow users at any time to start their call just by pushing the call button[4,5]. So with the help of the SDP over IMS architecture, we deploy the PoC service which is based on the functionalities of IMS and the services of SDP.

In this paper, we have implemented the PoC service deployment by utilizing the Java programming using JAIN Session Initiation Protocol(SIP) APIs. And JAIN SIP is a standard Java APIs to SIP for desktop and server applications[7]. After the studying the OMA PoC architecture and its functionalities, we want to realize it by implementing the prototype of PoC service. These achievements show us the feasibility of the PoC deployment and realization.

2. PoC Architecture and Protocol

The Open Mobile Alliance(OMA) specifies PoC as an IMS service using IMS internetwork interfaces to assure interoperability to other PoC networks, which will lead to rapid deployment of IMS technology by the telecommunication companies and operators[2].

The IMS Core provides basic IMS routing functionalities including the P(Proxy)-Call Session Control Function(CSCF), I(Interrogating)-CSCF, S(Serving)-CSCF and HSS(Home Subscriber Server). The reason for PoC service deployment into the IMS is that the PoC server itself does not need to handle issues such as Routing SIP signaling, Discovery and Address resolution services, SIP Compression, Charging, Authentication & Authorization, Accounting, Quality of Service(QoS) Control, and so on. And these functionalities are supported by IMS Core.

The Figure 1 describes the functional entities and the reference interfaces involved in the support of the PoC service in the SDP over IMS. More detail about Presence service can be referenced in [1].

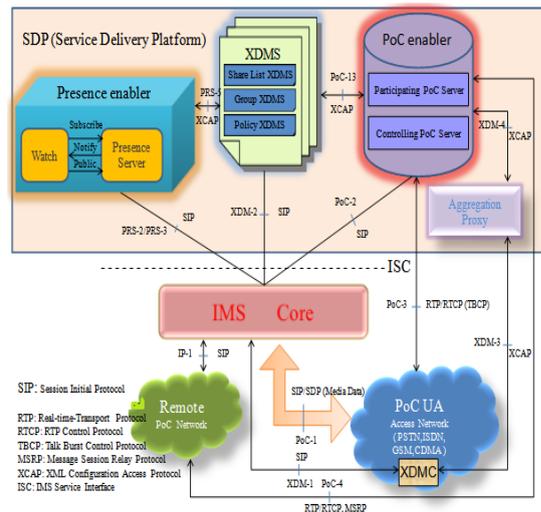


Fig. 1. PoC Architecture in the SDP over IMS

The involvement of the Presence enabler in OMA PoC architecture is to complement the push to talk service by allowing users to instantly see the

availability and willingness of other users to communicate. And it is tightly coupled with the PoC enabler. And the PoC architecture also utilizes the XML Document Management(XDM) enabler(XDMC/XDMS) to manage XML documents like group lists, access control lists, presence lists and Universal Resource Identifier(URI) lists which are created by the user and stored by the service provider by means of specific XML documents. The XDMS is an XML Configuration Access Protocol(XCAP) Server that can be considered as means of application configuration setting management as it stores application-specific configuration information. And it also notifies subscribers of PoC-specific document changes which are stored in the network. The Aggregation Proxy acts as the single contact point for the XDMC. The related definitions of the interfaces in PoC architecture are referenced in [2].

The IMS provides a service architecture on which applications can be deployed through the IMS Service Control(ISC) interface that can be represented as SIP. Also, it is common interface that implements filters for subscribers which have stored in the Home Subscriber Server(HSS). Since the SDP is a common platform that provides kinds of services, the ISC between SDP and IMS is such an interface that compares the filters with incoming SIP messages and determines which application services should be invoked[3]. Trigger points in initial Filter Criteria(iFC) are matched with information in the initial SIP request for instance, a specific SIP header, to determine which application servers(e.g., the PoC Server) are to be included and in which order in the SIP chain.

Considering using IMS Access Network solution, there is an another advantage of PoC service by embedding the PoC client into both the mobile phones and PCs that users could not only talk by cellular access but also by a fixed connection with wild coverage area.

Another importance of PoC service needed to be mentioned about is its media control for sessions. In

PoC, session control and other signaling are based on SIP, while voice traffic is carried through a Real-time Transport Protocol/Real-time Transport Control Protocol (RTP/RTCP) based streaming bearer. Among them SIP is used for session set up and RTP/RTCP are used for voice data transfer and floor control, respectively[4].

For the implementation of PoC service, there are SIP and RTP stacks listening out for data belonging to their respective user agents and sessions. If the packet is for a SIP user agent, the packet is fed into SIP stack and processed. If the data is RTCP data for a RTP session, the packet is fed into the RTP stack and processed. And if the packet is for a RTP session, the TCP/UDP packet is passed to the RTP stack for processing.

As PoC is a half-duplex communication, its under-control implementation is owed to Talk Burst Control Protocol(TBCP) or Media Burst Control Protocol(MBCP) that are used to arbitrate request from the PoC clients for the right to send media - in other words, floor control[5].

3. PoC Implementation Using JAIN SIP

3.1 Abstract of JAIN SIP

After giving an overview of the PoC architecture and protocols, we begin our research on realization of PoC by using Java programming. But firstly we need to describe how the JAIN SIP APIs works.

JAIN SIP is a standard Java APIs to the SIP for desktop and server applications. To understand JAIN SIP APIs much better, we depict the following architecture of JAIN SIP and each function of its entities and present them on the Figure 2 to show how Java components are arranged in JAIN SIP framework[6].

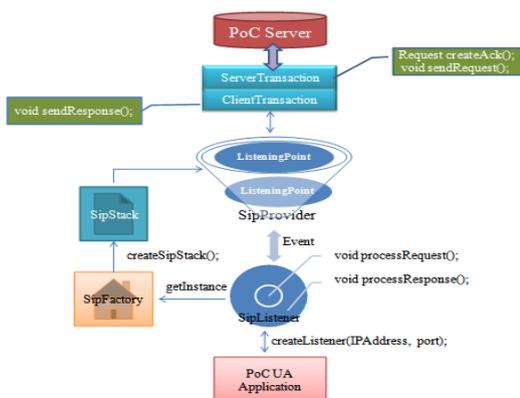


Fig. 2. JAIN SIP Object Architecture

The aim of JAIN SIP APIs is to allow developers to create services using high-level methods that are independent of the underlying network technology. With the JAIN SIP, programmers can implement the SIP elements more easily by fulfilling the processes and operations according to the SIP specification. Moreover, by applying other APIs, different features could be combined together to create a comprehensive program, which may be capable to manage multiple tasks. For instance, with the support of RTP and RTCP, SIP-based PoC service is capable of managing sessions and transfer multimedia.

JAIN SIP enables transaction stateless, transaction stateful and dialog stateful control over the protocol. JAIN SIP models both Client and Server Transaction as interfaces. When a request is sent out statefully, application must request a ClientTransaction and the stack associates it to a suitable ServerTransaction[6]. When the JAIN stack receives a response, it can associate a previously created ClientTransaction with the response. JAIN SIP uses Dialog for association between end-to-end users and maintains data needed for further message transmissions. Dialogs are established by creating Transactions such as INVITE and SUBSCRIBE and are managed by the stack.

3.2 Session Initiation

In the prototype of implementation, we integrate the IMS Core network into a component named ptt-proxy

shown in Figure 3, to supply routing, authentication and registration functions.

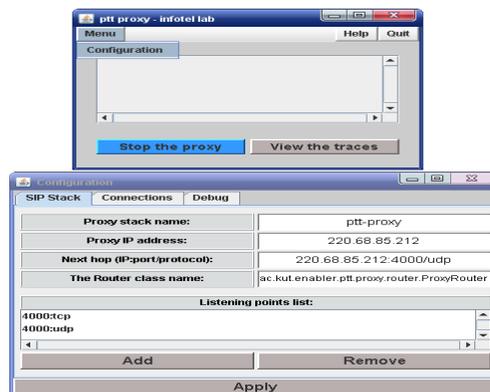


Fig. 3. A configuration of ptt proxy

To initiate a PoC session request in the JAIN SIP APIs, there are several ways to construct SIP request messages. The most direct way is to create all necessary headers and assemble them into a request by applying methods:

```
protected Request createRequest (requestName,
    callee, caller);
{
    //requestURI, callIdHeader, cSeqHeader, fromAddress,
    //toAddress, viaHeader, maxForwardHeader, contactHeader
}
```

For statefully forwarding the request, we should set the parameter "statefullForwarding" into "true" or "false" (by statelessly forwarding) in the method: requestForwarding.forwardRequest(targetURIList, sipProvider, request, serverTransaction, true);

Figure 4 describes the procedure in that the PoC User Agent(UA) registers on PoC server more visually and apprehensible. In this environment, we firstly construct the java method named "createRequest" used in the package of javax.sip.message.Request. After the "createRequest(Request.REGISTER, userSipURI, userSipURI)" has been finished, we need to create the method "clientTransaction.sendRequest()"

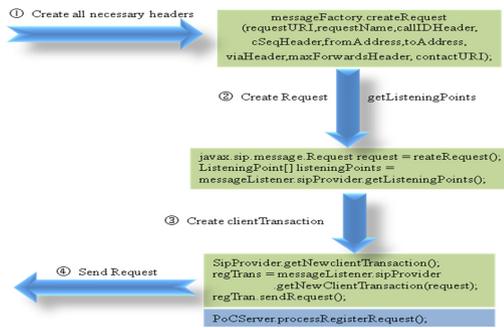


Fig. 4. Process of SIP Register Message

Because a Dialog is initialized by an INVITE request in order to further run the media session between UAs. So the user presses call button to send an INVITE request to other users and waits for acceptance of the callee(s). Figure 5 shows up the foundation of this Invite process to make one-to-one PoC session.

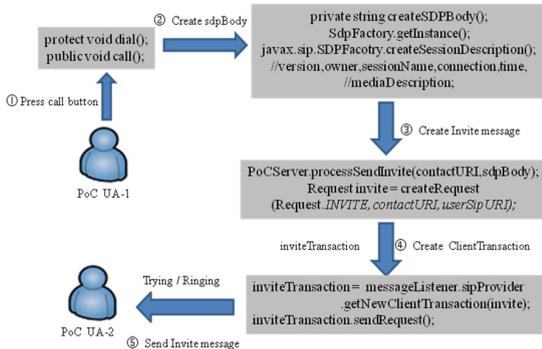


Fig. 5. Process of SIP Invite Message

4. PoC Prototype and Results

The prototype includes a graphic service creation environment that is based on Eclipse, an open-source integrated development environment(IDE) and JDK version 1.6 or later(J2SElibrary), JMF (JavaMediaFramework2.1.1.e) and JainSipApi1.2.jar for SIP interfaces and classes, JainSipRi1.2.jar for SIP reference implementation, log4j-1.2.8.jar for logging service, concurrent.jar for concurrency utilities, tracesviewer.jar for viewing the exchanges of SIP

messages and so on.

Our PoC service basically works like a walkie-talkie, while only one participant at a time is allowed to speak. After the callee has accepted the coming call, the session is able to be set up. The caller can now talk to the recipients for as long as he holds the speak-button. Under the control of the PoC Server and the TBCP/MBCP control protocol, the speaker with a highest priority is only allowed to speak until the one releases the button.

Figure 6 shows up the GUI view of one-to-one session foundation between the UAs.

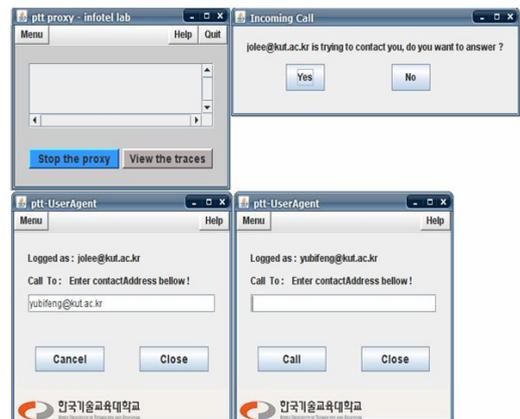


Fig. 6. A snapshot of Call Session

Note that PC Client may have the problem that it does not easily work with Network Address Translation(NAT). NAT is commonly used in home and office routers providing sharing of IP addresses among multiple users. SIP and RTP protocols, which are used in real-time communications, are not fully compatible with NAT. This also causes problems with PoC. The NAT problem could have been overcome with static IP addresses [9] and using the subscriber information stored in ptt-proxy server.

5. Conclusions

We have presented the advantages of PoC service

and PoC architecture. And we focus on the its practical implementation for the prototype that has been accomplished by using Java programming to show its capability and convenience.

The key benefits derived by this solution is that we could enhance the PoC service with more media type supporting which is called as Push-to-X service having the capability of using pictures, real-time video, and file transfer services remained to our future implementation work.

References

- [1] Jae-Hyoung Cho, Jae-Oh Lee, Seung-Chan Kang, "The Implementation of Presence Server in the M2M Using the RLS", *Journal of Next Generation Information Technology*, Vol 4. No. 8, pp 484-490, Oct., 2013.
- [2] Open Mobile Alliance, *Push to talk over Cellular (PoC) - Architecture Version 2.1*, Aug., 2011.
- [3] The 3rd Generation Partnership Project, 3GPPTS23.228v5.13.0, *IP Multimedia Subsystem*, Stage 2, Dec 2004.
- [4] Andrew Rebeiro-Hargrave & David Viamonte Sole, *Multimedia Group Communication*, John Wiley & Sons, Ltd., 2008.
DOI: <http://dx.doi.org/10.1002/9780470060476>
- [5] Open Mobile Alliance, *PoC User Plane Version 2.1*, Aug., 2011.
- [6] Phelim O'Doherty, Sun Microsystems, Mudumabai Ranganathan, NIST, *JAIN SIP Tutorial Serving the Developer Community*, 2003.
- [7] RFC 3261, *SIP: Session Initiation Protocol* Chapter 17, June 2002.
- [8] John Wiley & Sons LTD, *The IMS IP Multimedia Concepts and Services second edition*, 2006.
- [9] Helsinki University Of Technology, Elina Ahonen, *Implementing a Multi-access Push to Talk System and Analysing User Experience*, October 16, 2007.

Jae-Oh Lee

[Regular member]



- Feb. 1993 : Kwangwoon Univ., Dept. of Computer Science, PhD.
- March 1994 ~ Feb. 1995 : Kolon IC Research Center, Manager
- Feb. 1995 ~ Dec. 2000 : KT, Senior Researcher
- Sept. 1999 ~ Aug. 2002 : WarePlus Ltd., Vice President
- Aug. 2002 ~ current : Korea University of Technology and Education, School of Electrical, Electronics and Communication Eng., Professor

<Research Interests>

Network Control/Management, IoT, SDN/NFV

Hong-Kyu Lee

[Regular member]



- Feb. 1977 : Seoul National Univ., Dept. of Electronics Eng. BS
- Feb. 1979 : Seoul National Univ., Dept. of Electronics Eng. MS
- Aug. 1989 : Seoul National Univ., Dept. of Electronics Eng. PhD
- Mar. 1979 ~ Feb. 1992 : Agency for Defence Development, Senior Researcher
- Mar. 1992 ~ Current : Korea University of Technology and Education, School of Electrical, Electronics and Communication Eng., Professor

<Research Interests>

Genetic Algorithm, Moving Robot, Network Optimization