IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 11, Issue 5 (May. - Jun. 2013), PP 12-20
www.iosrjournals.org

Implementing Automatic Callback Using Session Initiation Protocol

Lisha Singh[1], Ashima Singh[2]
[1](*Assistant Professor, HMR institute of Technology & Management, GGSIP University, Delhi, India)*
[2](*Software Engineer,Aricent Group, Gurgaon, India)*

Abstract-*TheSession Initiation Protocol (SIP), is an application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. SIP is developed by SIP working group specified by the Internet Engineering Task Force (IETF), peer-to-peer communication protocol to establish, manipulate, and tear down communication sessions at application layer. These sessions include Internet telephone calls, multimedia distribution, and multimedia conferences. SIP invitations used to create sessions, carry session descriptions that allow participants to agree on a set of compatible media types. The Automatic Callback feature sends a priority ring on phone, indicating the previously busy or unanswered extension you dialed is now available. Callback feature allows to receive Callback notification when a called party line becomes available. Callback provides the ability of the system to suspend the call completion service if the user, who originated Callback, is currently busy and receives call-back notification when the called party becomes available. When the originating user then becomes available, the call completion service resumes for that user.*
Keywords-*Automatic Callback (ACB), Internet Engineering Task Force (IETF),Public Switched Telephone Network (PSTN), Session Initiation Protocol (SIP), Voice over Internet Protocol (VoIP)*

I. Introduction

There are many applications of the Internet that require the creation and management of a session, wherea session is considered an exchange of data between an association of participants. The implementation ofthese applications is complicated by the practices of participants: users may move between endpoints, theymay be addressable by multiple names, and they may communicate in several different media – sometimessimultaneously. Numerous protocols have been authored that carry various forms of real-time multimediasession data such as voice, video, or text messages. The Session Initiation Protocol (SIP) works in concertwith these protocols by enabling Internet endpoints (called user agents) to discover one another and toagree on a characterization of a session they would like to share. SIP is a signalingcommunications protocol, widely used for controlling multimediacommunication sessions such as voice and video calls over Internet Protocol (IP) networks. For locating prospective session participants,and for other functions, SIP enables the creation of an infrastructure of network hosts (called proxyservers) to which user agents can send registrations, invitations to sessions, and other requests. SIP makes use of elements called proxy servers to help route requests to the user's current location, authenticate and authorize users for services, implement provider call-routing policies, and provide features to users. SIP also provides a registration function that allows users to upload their current locations for use by proxy servers. SIP runs on top of several different transport protocols. SIP is anagile, general-purpose tool for creating, modifying, and terminating sessions that works independently ofunderlying transport protocols and without dependency on the type of session that is being established. Most services that exist in the public switched telephone network (PSTN) can be implemented trivially using SIP, either by itself or with currently proposed extensions. An efficient implementation of the Automatic Callback (ACB) service, however, cannot be accomplished without a mechanism for notification when a called party has become free. Typically, the caller initiates a call in the normal fashion for their access device (e.g. dialing a number, selecting an entry from an address book application). When the user receives indication of "ringing" or "busy," they may choose to activate the ACB service.

II. Sip Functionality

SIP is an application-layer control protocol that can establish, modify, and terminate multimedia sessions(conferences) such as Internet telephony calls. SIP can also invite participants to already existing sessions,such as multicast conferences. Media can be added to (and removed from) an existing session. SIP transparentlysupports name mapping and redirection services, which supports personal mobility - users canmaintain a single externally visible identifier regardless of their network location.

SIP supports five facets of establishing and terminating multimedia communications:
User location: determination of the end system to be used for communication;
User availability: determination of the willingness of the called party to engage in communications;
User capabilities: determination of the media and media parameters to be used;
Session setup: "ringing", establishment of session parameters at both called and calling party;
Session management: including transfer and termination of sessions, modifying session parameters,and invoking services.

SIP is not a vertically integrated communications system. SIP is rather a component that can be used withother protocols to build a complete multimedia architecture. Typically, these architectures will includeprotocols such as the Real-time Transport Protocol (RTP) for transporting real-timedata and providing QoS feedback, the Real-Time streaming protocol (RTSP) for controllingdelivery of streaming media, the Media Gateway Control Protocol (MEGACO) forcontrolling gateways to the PSTN, and the Session DescriptionProtocol (SDP) for describing multimedia sessions. Therefore, SIP should be used inconjunction with other protocols in order to provide complete services to the users. However, the basicfunctionality and operation of SIP does not depend on any of these protocols.SIP does not provide services. Rather, SIP provides primitives that can be used to implement differentservices. For example, SIP can locate a user and deliver an opaque object to his current location. If thisprimitive is used to deliver a session description written in SDP, for instance, the endpoints can agree on theparameters of a session. If the same primitive is used to deliver a photo of the caller as well as the sessiondescription, a "caller ID" service can be easily implemented. A single primitive istypically used to provide several different services.SIP does not offer conference control services such as floor control or voting and does not prescribe how aconference is to be managed. SIP can be used to initiate a session that uses some other conference controlprotocol. Since SIP messages and the sessions they establish can pass through entirely different networks,SIP cannot and does not provide any kind of network resource reservation capabilities. The nature of the services provided make security particularly important. To that end, SIP provides asuite of security services, which include denial-of-service prevention, authentication (both user to user andproxy to user), integrity protection, and encryption and privacy services.SIP works with both IPv4 and IPv6.

III. Automatic Callback

The Automatic Callback (ACB) feature allows to receive call-back notification when a called party line becomes available. One can activate Callback for a destination phone that is within the same Communications Manager cluster as your phone or on a remote PINX over QSIG trunks or QSIG-enabled intercluster trunks.To receive call-back notification, a user presses the Callback softkey while receiving a busy or ringback tone. A user can also activate Callback during reorder tone, which is triggered when the no answer timer expires. The calling phone only supports one active Callback request. The called phone can support multiple Callback requests. If the originating side (User A) gets reset after Callback has been activated, then Callback gets automatically cancelled. User A does not receive an audio alert, and the Callback notification screen does not display. If the terminating side (User B) gets reset, Callback does not get cancelled. User A will receive an audio alert, and the Callback notification screen displays after User B becomes available.
The following examples describe how Callback works after an unavailable phone becomes available:

1.User A calls User B, who is not available

User A calls User B, who exists either in the same Communications Manager cluster asUser A or in a different cluster. Because User B is busy or does not reply, User A activates the Callbackfeature by using the Callbacksoftkey. The following Callback activation message displays on the phoneof User A:

CallBack is activated on <DN of User B>
Press Cancel to deactivate
Press Exit to quit this screen

User A presses the Exit softkey
After User B becomes available (phone becomes on hook after busy or completes an off-hook andon-hook cycle from idle), User A receives an audio alert, and the following message displays on thephone of User A:

<DN of User B> has become available
Time HH:MM MM/DD/YYYY
Press Dial to call
Press Cancel to deactivate
Press Exit to quit this screen

User A presses the Exit softkey and then goes off hook and dials the DN of User B. User B answers thecall. Users A and B go on hook.When User A presses the Callbacksoftkey, the following message displays on the phone of User A:

<DN of User B> has become available
Time HH:MM MM/DD/YYYY
Press Dial to call
Press Cancel to deactivate
Press Exit to quit this screen

2.User A activates the Callback feature for User B but is busy when User B becomes available
User A calls User B. User B does not answer. User A activates the Callback feature by using theCallbacksoftkey. The following Callback activation message displays on the phone of User A:

CallBack is activated on <DN of User B>
Press Cancel to deactivate
Press Exit to quit this screen

User A presses the Exit softkey.
User C then calls User A, and users A and C go on hook in an active call. User B becomes available(phone becomes on hook after busy or completes an off-hook and on-hook cycle from idle) while UserA is still on an active call. User A receives an audio alert, and the following message displays on thephone of User A:

<DN of User B> has become available
Time HH:MM MM/DD/YYYY
Press Dial to call
Press Cancel to deactivate
Press Exit to quit this screen

User A can interrupt the active call to contact User B in either of two ways:
• Select Dial from the notification screen. The active call automatically gets put on hold while UserA calls User B.
• Press the Exit softkey to exit the notification screen and then park (or otherwise handle) the activecall. After the active call is handled, User A can press the Callbacksoftkey and select Dial to callUser B.

3.User A calls User B, who configured Call Forward No Answer (CFNA) to User C before call-backactivation occurs
The following scenario applies to Call Forward No Answer:
The call from User A gets forwarded to User C because Call Forward No Answer is configured for UserB. User A uses Callback to contact User C if User C is not busy; if User C is busy, User A contacts UserB.When User B or User C becomes available (on hook), User A receives an audio alert, and a messagedisplays on User A phone that states that the user is available.

4.User A calls User B, who configures call forwarding to User C after User A activates Callback
• User A calls User B, who exists in the same Communications Manager cluster as UserA. User A activates Callback because User B is not available. Before User B becomes available toUser A, User B sets up call forwarding to User C. User A may Callback User B or User C, dependingon the call-forwarding settings for User B.

• User A calls User B, who exists in a different cluster. The call connects by using a QSIG trunk. UserA activates Callback because User B is not available. Before User B becomes available to User A,User B sets up call-forwarding to User C. One of the following events occurs:

-If the Callback Recall Timer (T3) has not expired, User A always calls back User B

-After the Callback Recall Timer (T3) expires, User A may Callback User B or User C,depending on the call-forwarding settings of User B.

5. User A and User C call User B at the same time

User A and User C call User B at the same time, and User A and User C activate Callback because UserB is unavailable. A call-back activation message displays on the phones of User A and User C.When User B becomes available, both User A and User C receive an audio alert, and a message displayson both phones that states that User B is available. The User, that is, User A or User C, that presses theDial softkey first connects to User B.

IV. Suspend/Resume Functionality For Callback

Callback provides the ability of the system to suspend the call completion service if the user, whooriginated Callback, is currently busy and receives call-back notification when the called party becomesavailable. When the originating user then becomes available, the call completion service resumes for thatuser. After the originating user (User A) activates the Callback feature, and then becomes busy when thecalled party (User B) becomes available, the originating PINX sends out a Suspend Callback APDU message that indicates to the peer to suspend monitoring of User B until User A becomes available again. When User A becomes available, the originating PINX sends the Resume APDU message for theterminating side to start monitoring User B again.

The following example describes how the Suspend/Resume feature works:

• *User A is busy when User B becomes available*

User A calls User B, who exists either in the same Communications Manager cluster asUser A or in a different cluster. Because User B is busy or does not reply, User A activates the Callback feature by using the Callback softkey. The following Callback activation message displays on the phone of User A:

CallBack is activated on <DN of User B>
Press Cancel to deactivate
Press Exit to quit this screen

User A presses the Exit softkey.
User A has a busy trigger set to 1.
User A becomes busy. User B then becomes available.
User A does not receive an audio alert and does not receive a call-back notification screen on the display.
The originating side (User A) sends a Suspend Callback APDU message to the terminating side (User B).
User A becomes available. The originating side sends a Resume Callback APDU message to theterminating side. This causes monitoring of User B to resume.
When User B becomes available, User A receives an audio alert, and a Callback notification screendisplays.

V. System Requirementsfor Callback

Callback requires the following software components:

• Communications Manager 5.0 or later
• CallManager service that is running on at least one server in the cluster
• Database Layer Monitor service that is running on the same server as theCallManagerservice
• RIS Data Collector service that is running on the same server as the CallManagerservice
• Communications Manager Locale Installer, that is, if you want to use non-Englishphone locales or country-specific tones
• Microsoft Internet Explorer 7 or Microsoft Internet Explorer 8 or Firefox 3.x or Safari 4.x

VI. Implementation Of Automatic Callback

A variety of applications are enabled through knowledge of INVITE dialog usage state. Best example is Automatic Callback. ACBservice is sometimes referred to as "camp on extension," "call again," "automatic redial," and "automatic recall."Fig.1.describes a proposed implementation of an ACB service using SIP.

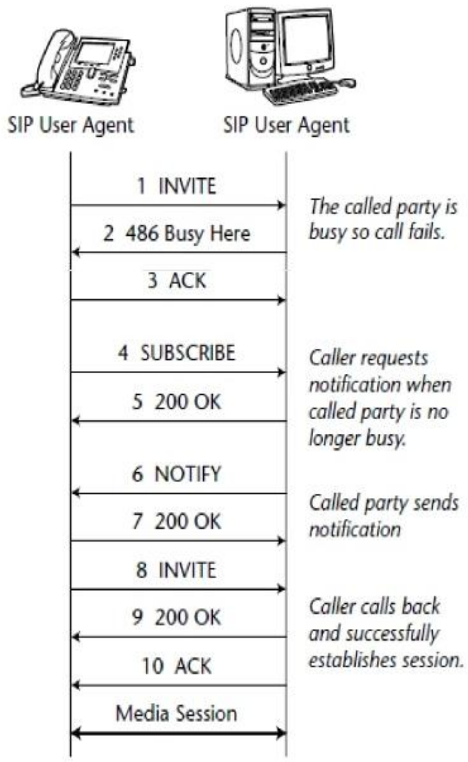

Fig. 1. Implementation of Automatic Callback

In this basic PSTN application, user A calls user B but User B isbusy. User A would like to get a callback when user B hangs up. When B hangs up, user A's phone rings. When A picks up,they hear ringing, while they are being connected to B. To implement this with SIP, a mechanism is required for A to receive a notification when the dialogs at B are complete. The method of activation will vary depending on the access device. For a hardware phone, there will typically be a dedicated "callback" button, PC-based clients may have a menu option and/or a hot key; and PSTN bridges may play a series of prompts to guide the user in activating the service. The user receives confirmation of service activation. Once the callee's device becomes available, the caller's phone will indicate that the ACB service has been triggered. For hardware phones and PSTN bridges, this will typically be a distinctive ring; for PC-based clients, it may be an audio alert accompanied by a dialogue box. The user responds to this alerting as if they were answering an incoming call. Upon doing so, user receives indication of the far end alerting, and the call proceeds as normal.

*1. **Requesting notification with SUBSCRIBE (callee busy)***

INVITE sip:a@ws592.abc.com SIP/2.0
To: A<sip:Auser@abc.com>
From: B<sip:Buser@xyz.org>; tag=38bec5a6-0000000a
Call-Id: 34a6-38bec5a6-61103dd1-3@terminal17.isp.net
CSeq: 5231 INVITE
Contact: sip:b@terminal17.isp.net
Content-Type: application/sdp

ACK sip:a@ws592.abc.com SIP/2.0
To: A<sip:Auser@abc.com>; tag=638bec5a67589416288
From: B<sip:Buser@xyz.org>; tag=38bec5a6-0000000a
Call-Id: 34a6-38bec5a6-61103dd1-3@terminal17.isp.net
CSeq: 5231 ACK
Content-Length: 0

Since this is a non-call-related subscription, the Call-ID is unique and does not match the previous INVITE request

SUBSCRIBE sip:a@ws592.abc.com SIP/2.0
To: A<sip:Auser@abc.com>
From: B<sip:Buser@xyz.org>;   tag=8a553c7c-38bebcb1
Call-Id: 3465-38bebcb1-5e86e396-6@terminal17.isp.net
CSeq: 16939 SUBSCRIBE
Contact: sip:b@terminal17.isp.net
Expires: 14400
Event: terminal-free
Content-Length: 0

SIP/2.0 200 OK
To: A<sip:Auser@abc.com>;   tag=346538bebcb150e
From: B<sip:Buser@xyz.org>; tag=8a553c7c-38bebcb1
Call-Id: 3465-38bebcb1-5e86e396-6@terminal17.isp.net
CSeq: 16939 SUBSCRIBE
Expires: 3600
Content-Length: 0

2. Requesting notification with SUBSCRIBE (no answer)

This call flow demonstrates the flow of messages if a user decides to activate the ACB service after the callee's terminal has started alerting, but before the callee answers.

INVITE sip:a@ws592.abc.com SIP/2.0
To: A<sip:Auser@abc.com>
From: B<sip:Buser@xyz.org>; tag=38bec5a6-0000000a
Call-Id: 34a6-38bec5a6-61103dd1-3@terminal17.isp.net
CSeq: 5231 INVITE
Contact:sip:b@terminal17.isp.net
Content-Type: application/sdp

SIP/2.0 180 Ringing
To: A<sip:Auser@abc.com>; tag=638bec5a6758941
From: B<sip:Buser@xyz.org>; tag=38bec5a6-0000000a
Call-Id:34a6-38bec5a6-61103dd1-3@terminal17.isp.net
CSeq: 5231 INVITE
Content-Length: 0

Since this is a non-call-related subscription, the Call-ID is unique and does not match the previous INVITE request.

SUBSCRIBE sip:a@ws592.abc.com SIP/2.0
To: A<sip:Auser@abc.com>
From: B<sip:Buser@xyz.org>; tag=8a553c7c-38bebcb
Call-Id:3465-38bebcb1-5e86e396-6@terminal17.isp.net
CSeq: 16939 SUBSCRIBE
Contact:sip:b@terminal17.isp.net
Expires: 14400
Event: terminal-free
Content-Length: 0

SIP/2.0 200 OK
To: A<sip:Auser@abc.com>; tag=346538bebcb150e
From: B<sip:Buser@xyz.org>; tag=8a553c7c-38bebcb
Call-Id: 3465-38bebcb1-5e86e396-6@terminal17.isp.net
CSeq: 16939 SUBSCRIBE
Expires: 3600

```
Content-Length: 0
```

Now that the service is activated, the client can cancel the INVITE transaction.

```
CANCEL sip:a@ws592.abc.com SIP/2.0
To: A<sip:Auser@abc.com>
From: B<sip:Buser@xyz.org>; tag=38bec5a6-0000000a
Call-Id:34a6-38bec5a6-61103dd1-3@terminal17.isp.net
CSeq: 5231 CANCEL
Content-Length: 0
```

```
SIP/2.0 200 OK
To: A<sip:Auser@abc.com>
From: B<sip:Buser@xyz.org>; tag=38bec5a6-0000000a
Call-Id:34a6-38bec5a6-61103dd1-3@terminal17.isp.net
CSeq: 5231 CANCEL
Content-Length: 0
```

```
SIP/2.0 487 Call Canceled
To: A<sip:Auser@abc.com>; tag=638bec5a6758941
From: B<sip:Buser@xyz.org>; tag=38bec5a6-0000000a
Call-Id:34a6-38bec5a6-61103dd1-3@terminal17.isp.net
CSeq: 5231 INVITE
Content-Length: 0
```

```
ACK sip:a@ws592.abc.com SIP/2.0
To: A<sip:Auser@abc.com>; tag=638bec5a6758941
From: B<sip:Buser@xyz.org>; tag=38bec5a6-0000000a
Call-Id:34a6-38bec5a6-61103dd1-3@terminal17.isp.net
CSeq: 5231 ACK
Content-Length: 0
```

3. Conveying terminal-free status using NOTIFY

The NOTIFY request contains the same Call-ID as the SUBSCRIBE that asked for it.

```
NOTIFY sip:b@terminal17.isp.net SIP/2.0
To: Bsip:Buser@xyz.org
From: A<sip:Auser@abc.com>; tag=38bebe1c-4953
Call-Id:3465-38bebcb1-5e86e396-6@terminal17.isp.net
CSeq: 15739 NOTIFY
Contact:sip:a@ws592.abc.com
Event: terminal-free
Content-Length: 0
```

```
SIP/2.0 200 OK
To: B<sip:Buser@xyz.org>; tag=346e38bebe434931fb51f33
From: A<sip:Auser@abc.com>; tag=38bebe1c-4953
Call-Id:3465-38bebcb1-5e86e396-6@terminal17.isp.net
CSeq: 15739 NOTIFY
Content-Length: 0
```

Now, a brand new INVITE is issued to begin a call to the newly-freed terminal.

```
INVITE sip:a@ws592.abc.com SIP/2.0
To: A<sip:Auser@abc.com>
From: B<sip:Buser@xyz.org>; tag=38bec5a6-7265482
Call-Id:355e-38bed457-269136c2-6@terminal17.isp.net
CSeq: 17960 INVITE
Contact:sip:b@terminal17.isp.net
```

Content-Type: application/sdp

SIP/2.0 180 Ringing
To: A<sip:Auser@abc.com>; tag=55e38bed45700ca
From: B<sip:Buser@xyz.org>; tag=38bec5a6-7265482
Call-Id:355e-38bed457-269136c2-63@terminal17.isp.net
CSeq: 17960 INVITE
Content-Length: 0

The call now continues as normal

4. Cancelling a previous ACB request
SUBSCRIBE sip:a@ws592.abc.com SIP/2.0
To: A<sip:Auser@abc.com>
From: B<sip:Buser@xyz.org>; tag=8f21337c-38a8191
Call-Id:3465-38bebcb1-5e86e396-6@terminal17.isp.net
CSeq: 7837 SUBSCRIBE
Contact:sip:b@terminal17.isp.net
Expires: 0
Event: terminal-free
Content-Length: 0

SIP/2.0 200 OK
To: A<sip:Auser@abc.com>; tag=213985987f9872a
From: B<sip:Buser@xyz.org>; tag=8f21337c-38a8191
Call-Id:3465-38bebcb1-5e86e396-6@terminal17.isp.net
CSeq: 7837 SUBSCRIBE
Expires: 0
Content-Length: 0

VII. Feature interactions with call forward, divert and voice-messaging system

The following call states describe the expected behaviors for the calling party, that occur when Communications Manager Callback interacts with the Call Forward, iDivert, and voice-messaging system features. When a called party (Phone B) either forwards an incoming call by using Forward All, Forward Busy, or Forward No Answer; or diverts a call by using iDivert; to a voice-messaging system, the calling party (Phone A) can enter one of the following states with respect to the Callback feature:

- VM-Connected state: The call gets connected to voice-messaging system. The Callback softkey remains inactive on the calling party (Phone A) phone.
- Ring-Out state with the original called party: The voice-mail profile of the called party does not have a voice-mail pilot. The called party (Phone B) will see "Key Is Not Active" after pressing the iDivertsoftkey. The calling party (Phone A) should be able to activate Callback against the original called party (Phone B).
- Ring-Out state with voice-messaging system feature and voice-mail pilot number as the new called party: The call encounters either voice-messaging system failure or network failure. The called party (Phone B) will see "Temp Failure" after pressing iDivertsoftkey. The calling party (Phone A) cannot activate Callback against the original called party (Phone B) because the call context has the voice mail pilot number as the "new" called party.
- Ring-Out state with busy voice-mail port and voice-mail pilot number as the new called party: The call encounters busy voice-mail port. The called party (Phone B) will see "Busy" after pressing iDivertsoftkey. The calling party (Phone A) cannot activate Callback against the original called party (Phone B) because the call context has the voice mail pilot number as the "new" called party.

VIII. Conclusion

ACB is a service that makes SIP as one of the protocols that grows rapidly and covers all field of communication over the world. Today there exist both commercial and open source implementation of Session Initiation Protocol, hardware and software based. New applications like ACB are being added all the time that contributes in becoming Voice over Internet Protocol(VoIP) more and more popular. The protocol is still under development and new features are being added. The focus from the beginning was to provide a new dynamic protocol that was powerful but still simple and ACB is a very strong feature that completes it. ACB service has benefit as no hassle remembering and dialing numbers of the callee to which caller is trying to reach. It also saves time by automatically redialing the number when the calleebecomes available.

References

[1] SIP: Session Initiation Protocol, by J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, *RFC 3261*, June 2002.

[2] M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg, "SIP: Session Initiation Protocol", *RFC 2543*, XYZ; March 1999

[3] Session Initiation Protocol, KK Tan and HL Goh .P.13101314 Volume 2, *IEEE International conference on Information Technology*, 11:th14: th of December 2002

[4] IETF SIP Working Group; [Online]. Available*:* *http://www.ietf.org/html.charters/sip-charter.html*

[5] *SIP Communication for Dummies* by Lawrence Miller, CISSP, and Peter H. Gregory, CISA, CISSP Foreword by Alan B. Johnston

[6] Cisco Unified Communications Manager Features and Services Guide, Release 8.5(1), [Online].Available:*http://www.cisco.com/en/US/docs/voice_ip_comm/cucm/admin/8_5_1/ccmfeat/fscallbk.html*

[7] "SIP: Session Initiation Protocol Service Examples", by S. Donovan, *RFC 5359*, October 2008

[8] *Internet Communications using SIP: Delivering VOIP and Multimedia Services with Session Initiation Protocol*, Second edition by Henry Sinnreich, Alan B. Johnston, Published by Wiley Publishing, Inc .

[9] A.Roach, "Event Notification in SIP", Internet Draft ;[Online]. Available: <*draft-roach-sip-subcribe-notify-00.txt*>, XYZ; March 2000.Work in progress.

[10] "Session Initiation Protocol (SIP)-Specific Event Notification" by A.Roach, *RFC 3265*, June 2002

[11] Fredrik Fingal, PatrikGustavsson, "A SIP of IP Telephony",[Online].Available:*http://www.cs.columbia.edu/sip/drafts/Fing9902_SIP*

[12] Rosenberg, J., Schulzrinne, H., and R. Mahy, "An INVITE-Initiated Dialog Event Package for the Session Initiation Protocol (SIP)", *RFC 4235*, November 2005

[13] "Automatic Callback Service in SIP" by A. Roach., IETF Internet Draft, Work in Progress