Automatic Callback: An INVITE Dialog Usage State Application for the Session Initiation Protocol

Lisha Singh

Abstract- Session 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. A variety of applications are enabled through knowledge of INVITE dialog usage state. Best example is Automatic Callback (ACB). The ACB 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.

Index Terms- Automatic Callback (ACB), Internet Engineering Task Force (IETF), Public Switched Telephone Network (PSTN), Session Initiation Protocol (SIP), Voice over Internet Protocol (VoIP)

1 INTRODUCTION

There are many applications of the Internet that require the creation and management of a session, where a session is considered an exchange of data between an association of participants. The implementation of these applications is complicated by the practices of participants: users may move between endpoints, they may be addressable by multiple names, and they may communicate in several different media – sometimes simultaneously. Numerous protocols have been authored that carry various forms of real-time multimedia session data such as voice, video, or text messages. SIP works in concert with these protocols by enabling Internet endpoints (called user agents) to discover one another and to agree on a characterization of a session they would like to share. SIP is a signaling communications protocol, widely used for controlling multimedia communication sessions such as voice and video calls over Internet Protocol (IP) networks. 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 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. 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

2 AUTOMATIC CALLBACK METHOD FOR A PACKET ORIENTED NETWORK

In cases when a called participant is not registered on a packet oriented network, a call-back request made by a calling subscriber is transmitted automatically to a substitution device which is disposed in a network and assigned to the called subscriber. Preferably, the callback request is stored in a separate database. As soon as the called subscriber is registered in the network, the callback request is transmitted to the called subscriber. ITU-T (International Telecommunication Union) Recommendation H.450.9 describes a method for an implementation of the known “automatic callback” feature in packet-oriented networks according to the H.323 standard. Analogously to timeslot-oriented communication networks, a distinction is made here between two different cases:

1) automatic callback in the event of a "busy" destination subscriber (CCBS) and
2) automatic callback in the event of a "free" destination subscriber (CCNR).

A requirement of both variants is for the subscribers to be registered on the network via their respective terminals. Besides the two statuses "busy" and "free", in packet-oriented networks employing the Internet Protocol—IP for short—there is another status in which the subscriber is neither "busy" nor "free" but is "not registered". This is the case, for example, when an application implemented on the relevant terminal has not been launched or the terminal is inactive.

2.1 Summary Of The Invention

An object of the present invention is to specify a method by which the "automatic callback" feature can also be executed in packet-oriented networks in cases where a called subscriber is "not registered" on the network. In the event that a called subscriber—or the terminal assigned to the called subscriber—is not registered on the packet-oriented network, then according to the invention a deputizing device—frequently referred to in the relevant literature as a proxy—will assume the functions of the called terminal. The deputizing device is generally implemented as a relevant function in a data processing device. The deputizing device allows the calling subscriber to place an automatic callback request with the called subscriber. As soon as the called subscriber registers on the packet-oriented network from a terminal, the callback request will be transmitted to this terminal and so will be available for executing the automatic callback. In cases where the called subscriber—or the terminal assigned to the called subscriber—is registered on the network when a callback request is activated by the calling subscriber but de-registers from the network before the callback is executed, the callback request will automatically be stored in the network (IP-N). As soon as the called subscriber re-registers on the network, the callback request will be transmitted to the called subscriber and executed by that subscriber.

A major advantage of a method according to the invention is that it can easily be implemented in existing systems. A further advantage of a method according to the invention lies in the called subscriber's ability to register from any terminal in the packet-oriented network, with the callback request being transmitted to the terminal on which the called subscriber has registered so that mobility required for the subscribers can be provided easily. An advantage of embodiments of the invention is that old and thus generally no longer current requests are automatically removed from the system through the cancellation of callback requests on expiration of a period of time that can be pre-specified, thereby minimizing the load on the system due to monitoring of the callback requests.

2.2 DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG. 1 is a block diagram for schematically representing the main functional units involved in a method according to the invention. It shows a first and a second zone Z-A, Z-B which are interconnected via a packet-oriented network IP-N, for example a Local Area Network (LAN for short). Data is transmitted between the zones Z-A, Z-B over the packet-oriented network IP-N by the H.323 and H.225 standards known per se of the ITU (International Telecommunication Union). A zone Z-A, Z-B in each case comprises what is called a gatekeeper GK-A, GK-B and several devices assigned to this gatekeeper GK-A, GK-B. Shown by way of example for each zone Z-A, Z-B is a terminal EG-A, EG-B assigned to a subscriber Tln-A, Tln-B, a deputizing device ST-A, ST-B, and a database DB-A, DB-B. Several zones are in turn combined into what is called a domain (not shown). A terminal EG-A, EG-B can be, for example, what is called an 'IP-Phone' or a 'personal computer' with a suitable application running on the personal computer. A zone Z-A, Z-B in each case comprises what is called a gatekeeper GK-A, GK-B and several devices assigned to this gatekeeper GK-A, GK-B. Shown by way of example for each zone Z-A, Z-B is a terminal EG-A, EG-B assigned to a subscriber Tln-A, Tln-B, a deputizing device ST-A, ST-B, and a database DB-A, DB-B.

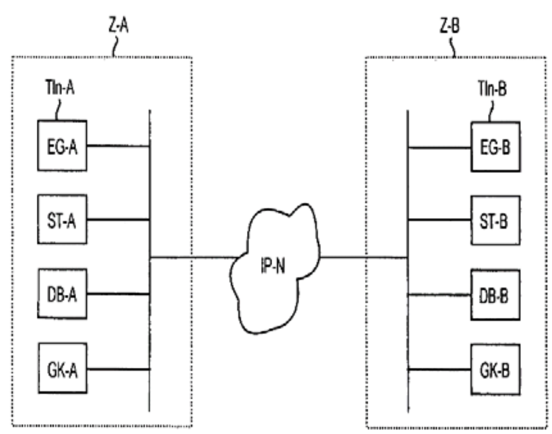

Fig. 1 Block Diagram of Main Functional Units

Several zones are in turn combined into what is called a domain (not shown). A terminal EG-A, EG-B can be, for example, what is called an 'IP-Phone' or a 'personal computer' with a suitable application running on the

ISSN 2229-5518

personal computer. Several terminals are usually assigned to a zone Z-A, Z-B, with the deputizing device ST-A, ST-B, the database DB-A, DB-B, and the gatekeeper GK-A, GK-B being assigned in this case jointly to the terminals. The deputizing device ST-A, ST-B, the database DB-A, DB-B and the gatekeeper GK-A, GK-B of a zone Z-A, Z-B can be either separate devices or a number of units combined into a single device. What is essentially performed by the gatekeeper GK-A, GK-B of a zone Z-A, Z-B is registration of the devices assigned to it and bi-directional address conversion between what are called alias addresses employed within a zone Z-A, Z-B and the addresses required by the packet-oriented network IP-N for connection setup. An alias address can be, for example, a telephone number or an e-mail address used by a subscriber Tln-A, Tln-B for a connection setup. Here the terminals EG-A, EG-B assigned to the subscribers Tln-A, Tln-B are in each case assigned to different zones Z-A, Z-B. The subscribers Tln-A, Tln-B or the terminals EG-A, EG-B can, however, also be assigned to a common zone Z-A, Z-B or a common domain.

A connection setup is initiated proceeding from a first subscriber Tln-A registered on the first gatekeeper GK-A of the first zone D-A to a second subscriber Tln-B of the second zone Z-B. This connection setup can be a voice connection, a video connection or a multimedia connection. The exemplary embodiment is based on the assumption that all calls to the second subscriber Tln-B are routed via the second gatekeeper GK-B. It is of no significance for this exemplary embodiment whether the calls from or to the first subscriber Tln-A are or are not routed via the first gatekeeper GK-A.

FIG. 2 shows a flowchart for elucidating the main messages that are exchanged by a method according to the invention. The unbroken thick arrows here represent a connection setup message for establishing a connection to a useful-data channel, the unbroken thin arrows represent a signaling message, the dot-dash arrows represent a message according to the known H.225.0 RAS protocol of the ITU, and the dotted arrows represent a database protocol message. In the present exemplary embodiment the first terminal EG-A sends a connection setup message “SETUP” to the second gatekeeper GK-B according to the H.323 standard. The first terminal EG-A can alternatively also transmit the connection setup message “SETUP” to the first gatekeeper GK-A, which forwards the connection setup message “SETUP” to the second gatekeeper GK-B. The second gatekeeper GK-B determines whether the second subscriber Tin-B is currently registered on the second gatekeeper GK-B. If this is not the case, the second gatekeeper GK-B will forward the received connection setup message “SETUP” to the second deputizing device ST-B. The second deputizing device ST-B then functions as a deputy for the second terminal EG-B and sends a message “ALERTING” according to the H.225.0 standard back to the first terminal EG-A.

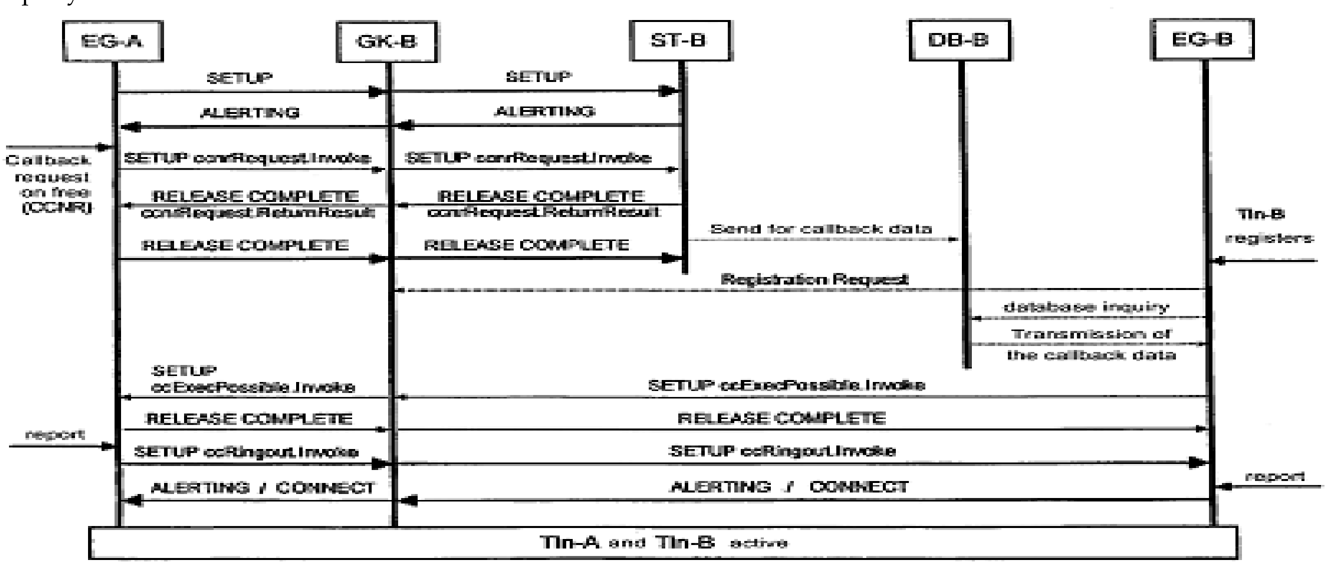

Fig. 2 Flow Chart of exchanged messages between Functional Units

On receipt of the message “ALERTING”, the first terminal EG-A is set to a status in which the feature “automatic callback on free” can be initiated on the first terminal EG-A according to the H.450.9 standard. For this, a new signaling connection containing the H.450.9 operation "ccnrRequest.Invoke" in a signaling message "SETUP" according to the H.225.0 standard is set up between the first terminal EG-A and the second gatekeeper GK-B. This signaling message "SETUP" is, as previously, forwarded from the second gatekeeper GK-B to the second deputizing device ST-B. The operation "ccnrRequest.Invoke" contains the directory number of the first subscriber Tln-A, the directory number of the second subscriber Tln-B and, optionally, a callback identifier and information about the required service type (voice, video, multimedia). This data will be referred to below as callback data. The second deputizing device ST-B accepts the callback request of the first subscriber Tln-A and sends an acknowledgement message "RELEASE COMPLETE" according to the H.225.0 standard together with an operation "ccnrRequest.ReturnResult" according to the H.450.9 standard back to the first terminal EG-A. On receipt of the acknowledgement message "RELEASE COMPLETE", the first terminal EG-A releases the still existing connection of the original connection setup in the alerting status with the message "RELEASE COMPLETE" according to the H.225.0 standard. In an ensuing operation, the second deputizing device ST-B transmits the callback data to the second database DB-B, in which the callback data is buffered. The second database DB-B can be physically located on the same device on which the second deputizing device ST-B has been implemented. Any database protocol, such as ODBC, JDBC, XML can be used as the interface and protocol between the second deputizing device ST-B and the second database DB-B.

If the second subscriber Tin-B registers on the second gatekeeper GK-B within a period of time that can be pre-specified—within 24 hours, for example—communication will take place between the second terminal EG-B and the second database DB-B. The period of time can be monitored in the first terminal EG-A by, for example, what is called an "SS-CC service duration timer" according to the H.450.9 standard. Alongside other information—the user profile, for example—the second database DB-B then also transmits information about any outstanding callbacks, i.e. the callback data, to the second terminal EG-B. The second terminal EG-B is then able to execute the callback procedure according to the H.450.9 standard. In the present exemplary embodiment this is effected by sending an availability operation "ccExecPossible.Invoke" to the first terminal EG-A as part of a signaling message "SETUP" according to the H.225.0 standard. It is assumed in the present exemplary embodiment that all calls to or from the second subscriber Tln-B are routed via the second gatekeeper GK-B. In the present exemplary embodiment the first terminal EG-A is registered on the first gatekeeper GK-A and has the status "free". The first subscriber Tln-A is called locally from the first terminal EG-A and the signaling connection between the first terminal EG-A and the second terminal EG-B is released from the first terminal EG-A by the signaling message "RELEASE COMPLETE". If the first subscriber Tln-A answers, this will result in the actual "automatic callback" from the first terminal EG-A to the second terminal EG-B in the form of a connection setup according to the H.323 and H.450.9 standards by a connection setup message "SETUP", which contains an H.450.9 operation "ccRingout.Invoke". If the second subscriber Tln-B answers, connection setup messages "ALERTING" and "CONNECT" according to the H.225.0 standard will be transmitted from the second terminal EG-B to the first terminal EG-A. The connection setup messages "SETUP, "ALERTING", and CONNECT" contain information (not shown) needed in order to set up the useful-data channels according to the H.323 standard.

Transmission of the callback data by the second deputizing device ST-B to a second database DB-B makes it possible for the second subscriber Tln-B to register on the second gatekeeper GK-B from any terminal and for the callback to proceed nonetheless to its further implementation. This is facilitated by transmission of the callback data stored in the second database DB-B to the terminal on which the second subscriber Tln-B registers. A method according to the invention thus also provides a required mobility for the subscribers Tln-A, Tln-B. According to a further embodiment of the invention, the second subscriber Tln-B is registered on the second gatekeeper GK-B when the callback request is activated by the first subscriber Tln-A. In this case, the callback data according to the H.450.9 standard will be stored not in the second deputizing device ST-B but in the second terminal EG-B. If, however, the callback is not executed because the second subscriber Tln-B has de-registered from the second gatekeeper GK-B, then proceeding from the second terminal EG-B the callback data will be transmitted to the second database DB-B. When the second subscriber Tln-B re-registers on the second gatekeeper GK-B, the callback data will be transmitted again to the terminal via which the second subscriber Tln-B has currently registered and the "automatic callback" will be executed as described above.

The first subscriber Tln-A is not registered on the first gatekeeper GK-A when the callback is executed. If this is the case, then the first deputizing device ST-A of the first domain D-A will, together with the first database DB-A, assume the functions of the first terminal EG-A. The first deputizing device ST-A will here respond to the availability operation "ccExecPossible.Invoke" of the second terminal

ISSN 2229-5518

EG-B instead of the first terminal EG-A with the operation "ccSuspend.Invoke" according to the H.450.9 standard. When the first subscriber Tln-A re-registers on the first gatekeeper GK-A, the original callback request will be reactivated by an operation "ccResume.Invoke". It is possible here to ask the first subscriber Tln-A, before the callback is executed, whether the callback is still required. If the first subscriber Tln-A de-registers, the deputizing device ST-A will also assume the function of callback time monitoring from the first terminal EG-A and cancel the callback request by an operation "ccCancel.Invoke" according to the H.450.9 standard if the period of time that can be pre-specified is exceeded.

3 AUTOMATIC CALLBACK

ACB feature allows to receive call-back notification when a called party line becomes available. The following examples describe how Callback works after an unavailable phone becomes available:

The following examples describe how Callback works after an unavailable phone becomes available:

3.1 User A calls User B, who is not available

User A calls User B, who exists either in the same Communications Manager cluster as User 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

After User B becomes available (phone becomes on hook after busy or completes an off-hook and on-hook cycle from idle), User A receives an audio alert, and 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

User A presses the Exit softkey and then goes off hook and dials the DN of User B. User B answers the call. Users A and B go on hook. When User A presses the Callback softkey, 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

3.2 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 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 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 User A is still on an active call. User A receives an audio alert, and 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

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 User A calls User B.
• Press the Exit softkey to exit the notification screen and then park (or otherwise handle) the active call. After the active call is handled, User A can press the Callback softkey and select Dial to call User B.

3.3 User A calls User B, who configured Call Forward No Answer (CFNA) to User C before call-back activation 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 User B. User A uses Callback to contact User C if User C is not busy; if User C is busy, User A contacts User B. When User B or User C becomes available (on hook), User A receives an audio alert, and a message displays on User A phone that states that the user is available.

3.4 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 User A. User A 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. User A may Callback User B or User C, depending on 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. User A 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.

3.5 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 User B 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 displays on both phones that states that User B is available. The User, that is, User A or User C, that presses the Dial softkey first connects to User B.

4 SUSPEND/RESUME FUNCTIONALITY FOR CALLBACK

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. After the originating user (User A) activates the Callback feature, and then becomes busy when the called 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 the terminating 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 as User 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 the terminating side. This causes monitoring of User B to resume.

When User B becomes available, User A receives an audio alert, and a Callback notification screen displays.

5 SYSTEM REQUIREMENTS FOR 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 the CallManager service
• RIS Data Collector service that is running on the same server as the CallManager service
• Communications Manager Locale Installer, that is, if you want to use non-English phone locales or country-specific tones
• Microsoft Internet Explorer 7 or Microsoft Internet Explorer 8 or Firefox 3.x or Safari 4.x

6 IMPLEMENTATION OF AUTOMATIC CALLBACK

A variety of applications are enabled through knowledge of INVITE dialog usage state. Best example is Automatic Callback. ACB service 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.

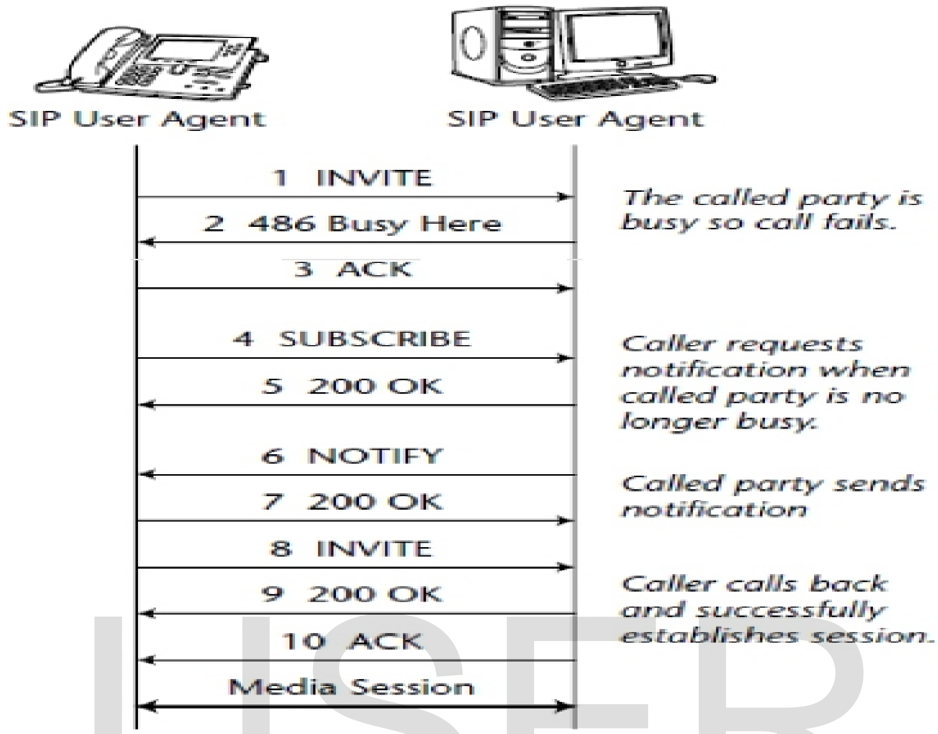

Fig. 1. Implementation of Automatic Callback

In this basic PSTN application, user A calls user B but User B is busy. 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.

6.1 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

SIP/2.0 486 Busy
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

ISSN 2229-5518

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

6.2 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

ISSN 2229-5518

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

6.3 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

6.4 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

7 FEATURE INTERACTIONS WITH CALL FORWARD, iDIVERT 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 iDivert softkey. 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 iDivert softkey. 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 iDivert softkey. 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.

8 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 features that completes it.

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

ISSN 2229-5518

[11] Fredrik Fingal, Patrik Gustavsson, "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