Implementing Bridged Line Appearances Using Session Initiation Protocol (SIP)

Abstract

This document describes a proposal for implementing Bridged Line Appearance for SIP User Agents.

Table of Contents

Status of this Memo

Abstract........................................................... 1
Introduction..................................................... 3
Terminology........................................................ 3
Conventions used in this document............................. 4
Feature Description/Overview.................................... 4
Usage Scenarios.................................................... 4
Overview of Operation............................................ 5
Call Flows.......................................................... 6
State Recovery.................................................... 10
Security Considerations.......................................... 10
References......................................................... 11
Acknowledgements................................................ 11

Anil Kumar

Internet-Draft Bridged Line Appearance 6/11/2003

Author’s Addresses............................................... 11
Full Copyright Statement........................................ 11
Introduction

Recently, the next generation voice market has seen the introduction of user agents that are equipped with more powerful processors, richer user interfaces, and the ability to provide end user features well beyond that of their predecessors. As these user agents evolve to track the pace of technology, so should the protocol itself.
evolve in order to take advantage of this new medium for providing a rich set of enhanced features to the end user. The ability of SIP to enable a richer user experience is essential for the adoption of SIP as the protocol of choice for business communications, and will accelerate the deployment of SIP user agents in the workplace.

Several draft documents exist that define and provide best current practices for implementing various single line telephony features using SIP, none exist that provide details for implementing business telephony related features. A few other Internet drafts provide high-level guidelines as to what features get implemented on a proxy as against a user agent and vice-versa, but don’t provide any specific guidelines.

This document details a proposal for implementing one of the most popular features expected of business phones – Bridged Line Appearance (BLA).


Terminology

Directory Number (DN): A telephone number or a SIP-URL assigned to a business phone.

Bridged Line Appearance (BLA): A DN that appears on more than one users business phone.

Single Call Arrangement (SCA): An arrangement where in only one business phone among a group of business phones that share a DN can be active at any point of time.

Line Key/Appearance: A button on a business phone faceplate used for outgoing and incoming calls

Primary Appearance: A user or a telephone that "owns" a particular DN.

BLA group: All users that share a BLA appearance on their telephone are referred to as belonging to a BLA group.

Event Agent (EA): A Event agent is a SIP user agent, which handles SUBSCRIBE requests and generates notifications of changes in dialog state. The Event Agent SUBSCRIPTES to registration events. As a result, the EA would learn about registration state of User Agents. EA then generates subscription to the User Agents for dialog package and receives notification of dialog state events from the User Agents. SIP URI uniquely identifies the Event Agent.

Conventions used in this document
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [1] and indicate requirement levels for compliant mechanisms.

Feature Description/Overview

BLA allows a single DN to be monitored by multiple user agents. When a call is made to this DN, the call is offered to all user agents that have mapping to this DN.

One user in the BLA group is allowed to answer the incoming call. Once the call is answered, the other users in the BLA group will be able to monitor the status of the call by means of visual indicators on their user agents.

If the user that answered the call is in conversation, the BLA group is displayed as being busy. No other user in the group may pick up that call.

If the user that answered the call places the dialog on hold, another member of the BLA group may pick up the call. This held state is visually indicated to the BLA group.

Users within the BLA group alter local settings on their user agents, such as how they are alerted if a bridged line is being called or how they are alerted if a bridged line is placed on hold. The user agents can be configured to play regular alerting tone, mute with a visual indicator, play a call waiting beep etc.

A BLA user may only place a call from a line associated with a BLA group if that group is idle. When the user places such a call, all members of the BLA group are notified of this usage, and are blocked from using this line appearance until the line appearance goes back to "Idle" state or if the media is placed on Hold.

Usage Scenarios

There are many applications of this feature:

- An assistant might have bridged line appearances of supervisor's extensions

- Business main numbers can be bridged on multiple phones & answered by group of users

- Work at more than one location in a huge campus like University, Hospital etc.

Overview of Operation

This proposal uses the following components to enable this feature:

- SIP Registrar
- SIP Forking Proxy
- SIP Event Agent (EA)

The Event Agent SUBSCRIBES to the registration event package as outlined in [5] for users belonging to a BLA group. Thus, it has knowledge of all users registered to a BLA group at any point of time.

All User Agents that belong to the BLA group MUST support the dialog state package as outlined in [4] to NOTIFY other User Agents of the dialog-state.

User Agents UA1, UA2 & UA3 belong to a BLA group and REGISTER for the same address-of-record (bridged1@tabasco.com).

The registrar NOTIFY’s the Event Agent of successful registration of each UA.

The Event Agent SUBSCRIBE to User Agents for the state of all Dialogs as defined in [4].
The User Agents SUBSCRIBE to Event Agent for the state of all Dialogs as defined in [4].

The User Agents act as the notifier for the dialog state events as defined in [4].

Forking Proxy forks INVITE.

The Event Agent sends NOTIFY of dialog state events to all the User Agents.

The User Agents in the group could SUBSCRIBE to each other & NOTIFY dialog state events, but in a large group the User Agents have to manage a larger number of SUBSCRIPTIONS & NOTIFICATIONS. The Event Agent helps in managing large groups better. Further, the Event User Agent MAY filter dialog state events & NOTIFY User Agents only the dialog state events which are required for the application or feature. The Event Agent MAY also SUBSCRIBE to dialog state events with "filters" to reduce the number of NOTIFY messages exchanged between the Event Agent and the UA’s in the BLA group.

Call Flows

Call Placed to a BLA extension

In the call flow below "User A" is the primary appearance. "User B" and "User C" have bridged appearances of "User A". "User X" places a call to "User A".

All NOTIFYÃs in the call flow below carry "dialog" events and only dialog states are mentioned for simplicity.

<table>
<thead>
<tr>
<th>User X</th>
<th>Proxy</th>
<th>User A</th>
<th>User B</th>
<th>User C</th>
<th>Event Agent</th>
</tr>
</thead>
</table>

Anil Kumar

Internet-Draft Bridged Line Appearance 6/11/2003

<table>
<thead>
<tr>
<th>(CID 1,frmtag=1000)</th>
<th>INVITE</th>
<th>--------&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 TRYING INVITE</td>
<td>---------&gt; INVITE</td>
<td></td>
</tr>
<tr>
<td>180 RINGING</td>
<td>&lt;--------</td>
<td></td>
</tr>
<tr>
<td>180 RINGING</td>
<td>180 RINGING</td>
<td>&lt;------------------</td>
</tr>
</tbody>
</table>

(CID 1,frmtag=1000, totag=2000)
Anil Kumar
Internet-Draft Bridged Line Appearance 6/11/2003

SDP=a:sendonly

INVITE (Replaces: Cid=1, from-tag=1000, totag=2000)

INVITE CID = 200

200 OK 200 OK

NOTIFY/200 OK (confirmed)

NOTIFY/200 OK (confirmed)

NOTIFY/200 OK (confirmed)

NOTIFY/200 OK (confirmed)

NOTIFY/200 OK (confirmed)
Anil Kumar
Internet-Draft Bridged Line Appearance 6/11/2003

Call Placed from a BLA extension

User X Proxy User A User B User C Event Agent

<table>
<thead>
<tr>
<th>INVITE</th>
<th>-------</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 TRYING</td>
<td></td>
</tr>
<tr>
<td>180 RINGING</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>180 RINGING</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>NOTIFY/200 OK</td>
<td>(trying)</td>
</tr>
<tr>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>NOTIFY/200 OK</td>
<td>(trying)</td>
</tr>
<tr>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>NOTIFY/200 OK</td>
<td>(trying)</td>
</tr>
<tr>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>NOTIFY/200 OK</td>
<td>(trying)</td>
</tr>
<tr>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>NOTIFY/200 OK</td>
<td>(terminated)</td>
</tr>
<tr>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>NOTIFY/200 OK</td>
<td>(terminated)</td>
</tr>
<tr>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>NOTIFY/200 OK</td>
<td>(terminated)</td>
</tr>
<tr>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>NOTIFY/200 OK</td>
<td>(terminated)</td>
</tr>
<tr>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>NOTIFY/200 OK</td>
<td>(terminated)</td>
</tr>
</tbody>
</table>
In this scenario, the UA just before allowing the user to place a call, shall send out a NOTIFY with event as "trying". On getting a 200 OK response, the UA shall allow the user to place the call (send out the INVITE). In case a UA receives a non-200 final response it MUST inspect the Retry-After interval & reinitiate the NOTIFY request. This is needed to disallow use of the BLA line appearance once one of the users has chosen the same.

Sending NOTIFY dialog state of "trying" before an INVITE

Anil Kumar
Internet-Draft Bridged Line Appearance 6/11/2003

is a departure from [4], but this MUST be implemented to resolve glare issues.

State Recovery

A critical aspect for reliable operation of this feature is the ability for all UA’s in the BLA group to recover from network failures caused at a single UA. For example, one of the user agents in the BLA group answered an incoming call, notifies the dialog state to other members and then experiences a network outage. The calling UA could have detected the same (using RTP or some other means) and could have hung up. However, none of the other UA’s in the BLA group gets notification of this change in dialog state and their BLA appearance could stay out of sync for a long time; depending on when the network is restored, or when the Event Agent attempts to refresh its dialog-state subscription with this UA. To recover from such a failure, the Event Agent MAY SUBSCRIBE with a shorter expiration (expiration interval no smaller than 300 seconds is RECOMMENDED) following the notification of a "confirmed" dialog with the UA’s that notified this information.

Security Considerations

Since many of these features are implemented using semantics provided by RFC 3261 [2], Event Package for Dialog State as define in [4], Multi-party application framework [6], and Event Notification [3], security considerations in these documents apply to this draft as well.

Specifically, since dialog state information and the dialog
identifiers are supplied by UA’s in a BLA group to other members, the same is prone to "call hijacks". For example, a rogue UA could snoop for these identifiers and send an INVITE with Replaces header containing these call details to take over the call. As such INVITES with Replaces header MUST be authenticated using the standard mechanism (like Digest or S/MIME) described in RFC 3261 [2] before it is accepted. NOTIFY message bodies that provide the dialog state information and the dialog identifiers MAY be encrypted end-to-end using the standard mechanics.

So also, All SUBSCRIBES between the UA’s and the Event Agent MUST be authenticated.

Reference


Acknowledgements

The authors would like to thank Kent Fritz, John Weald, Sunil Veluvalli, and Graeme Dollar Sylantro Systems for their comments and suggestions during authoring of this draft.

Author’s Addresses

Anil Kumar                  Email: anil.kumar@sylantro.com
sip:anil.kumar@sip.sylantro.com
(408) 626 2332

Venkatesh Venkataramanan    Email: venkatar@sylantro.com
sip:venkatar@sip.sylantro.com
(408) 626 3025

Full Copyright Statement

Copyright (C) The Internet Society (2003). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.