SIP
Internet-Draft
Expires: August 1, 2008

D.R. Worley
Bluesocket
January 29, 2008

Call Completion Implementation Details
draft-worley-call-completion-01

Status of this Memo

This Internet-Draft will expire on August 1, 2008.

Copyright Notice

Abstract

This paper gives a detailed discussion of the implementation of "call completion on busy subscriber" and "call completion on no reply" to flesh out the proposed call completion event package.

Table of Contents

1. Introduction

This paper is a companion to and extension of [1], in order to give a more detailed discussion of the implementation of the "call completion on busy subscriber" and "call completion on no response" features within SIP[2]. It is presented separately from [1] in order to allow this proposal to be discussed and refined before integrating it with the more mature work in [1]. This paper assumes a familiarity with the concept of call completion features, as well as familiarity with the implementation proposed in [1].

The call-completion architecture is driven by the interactions between two types of agents, a "caller's agent" which operates on behalf of the original caller, and a "callee's monitor", which operates on behalf of the original callee. In order to allow flexibility and innovation, most of the interaction between the caller's agent and the caller-user(s) and the caller's UA(s) is out of the scope of this document. Similarly, most of the interaction between the callee's monitor and the callee-user(s) and the callee's UA(s) is out of the scope of this document, as is also the policy by which the callee's monitor arbitrates between multiple call-completion requests. (Although simple agents and monitors can be devised that interact with users and UAs entirely through standard SIP mechanisms[2][3][4].)

2. Outline of the Call-Completion Mechanism

The call-completion architecture augments each caller's UA (or UAC) which wishes to be able to use the call-completion features with with a "call-completion agent" (also written as "CC agent", "agent", or "caller's agent"). It augments each callee's UA (or UAS) which wishes to be able to be the target of the call-completion features with a "call-completion monitor" (also written as "CC monitor", "monitor", or "callee's monitor"). The caller's agent subscribes to the call-completion event package[1] of the callee's monitor in order to coordinate with the monitor (and indirectly with other callees' monitors) to implement the call-completion features.

When the caller's UA makes a call to a callee that fails (e.g., because the callee was busy or the callee did not answer), and the caller wishes to use CC to contact the callee later, the caller instructs the caller's agent to activate the CC feature.

Note that SIP call-completion does not inherently distinguish "call completion on no reply" (CCNR) from "call completion on busy subscriber" (CCBS), because the network does not need to make a distinction, and given the potential complexity of SIP routing, agents in the network may not be able to.

The caller's agent sends a SUBSCRIBE request for the call-completion event package to the original destination URI of the call. This SUBSCRIBE reaches the callee's monitor. The callee's monitor uses the existence of the subscription to know that the caller is interested in using the CC feature in regard to the original call. The monitor keeps a list or queue of failed calls to the callee, and of the caller's agent subscriptions, which indicate the callers that are waiting to use the CC features.

When the callee's monitor judges that the user and/or user's UA is available for call-completion, the callee's monitor selects (usually) one caller's agent to be the next caller to execute call-completion to the callee. The callee's monitor sends a call-completion event update to the selected caller's agent telling it to begin execution of call-completion.

When the caller's agent receives this update, it calls the caller's UA or otherwise tests whether the caller is available to take advantage of call-completion. If the caller is available, the agent directs the caller's UA to make again the call to the callee. This call is identified as a call-completion call so it can be given precedence in reaching the callee's UA.

3. Detailed Description of the Call-Completion Mechanism

3.1. Caller's Call-Completion Agent

The call-completion architecture augments each caller's UA (or UAC) which wishes to be able to use the call-completion features with with a "call-completion agent". An agent may be associated with more than one UA if it is common that a caller or population of users will be shared between the UAs, and especially if the UAs share an AOR.

The caller's agent has a method of monitoring calls made from the UA(s) in order to determine their Call-Id's and (potentially) their final response statuses. This may be achieved by subscribing to the dialog event package of the UA(s) or by other means.

The callers using the UA(s) can indicate to the caller's agent when they wish to avail themselves of CC for a recently-made call which failed to reach their desired destination. This may be achieved by an INVITE to a special URI which is routed to the caller's agent or by other means.

The caller's agent has a method of monitoring or querying the state of the UA(s) and/or user(s) to determine when they are available to be used for a CC call. This may be achieved by subscribing to the dialog event package of the UA(s), by a user-initiated an INVITE to the UA(s), or by other means.

The caller's agent can order the UA(s) at which the relevant calling user(s) are available to generate a CC call to the callee. This may be achieved by sending a REFER to the UA(s) or by other means.

3.2. Callee's Call-Completion Monitor

The call-completion architecture augments each callee's UA (or UAS) which wishes to be able to be the target of the call-completion features with a "call-completion monitor". A monitor may be associated with more than one UA if it is common that a callee or population of callees will be shared between the UAs, and especially if the UAs share an AOR.

The callee's monitor has a method of monitoring calls made to the UA(s) in order to determine their Call-Id's and (potentially) their final response statuses. This may be achieved by subscribing to the dialog event package of the UA(s) or by other means, such as communication with the UA's "home proxy".

The callees using the UA(s) may be able to indicate to the callee's monitor when they wish to receive CC calls.

The callee's monitor has a method of monitoring the state of the UA(s) and/or user(s) to determine when they are available to receive a CC call. This monitoring may be achieved by subscribing to the dialog event package of the UA(s), by a callee-initiated INVITE to a special URI which is routed to the callee's monitor, or by other means. The criteria of availability may vary depending on information regarding the original calls of the current CC requests. E.g., if an original call failed because the UA was busy, the UA may be considered available for CC when it becomes not-busy, but if an original call failed because the UA was not answered, the UA may be considered available for CC when it becomes not-busy after being busy with an established call.

The callee's monitor maintains information about the set of INVITEs that have been received by the UA(s) that did not obtain successful final responses. In practice, the monitor may remove knowledge about an incoming dialog from its set if its CC policy establishes that the dialog is no longer eligible for CC.

3.3. The Original Call Is Made

The caller's UA sends an INVITE to a request URI. One or more forks of this request reach one or more of the callee's UAs. By hypothesis, none of the callee's UAs returns a success response, as otherwise, call completion services would not be needed for this call. However, the caller's INVITE might succeed at some other UA that the calling user considers insufficient to satisfy his needs. E.g., a call that is not answered by the callee user may connect to the callee user's voicemail server. Eventually, the INVITE fails, or the resulting dialog(s) are terminated. Note that the Call-Id of the INVITE is a unique identifier of this call attempt.

The caller's agent records the request URI, the Call-Id, and possibly other information. The callee's monitor records the Call-Id and possibly other information.

Note that the caller's UA may not receive any response from any of the callee's UA(s), as the final response returned to the caller's UA may have been from a fork that reached a UA that was not the callee's.

3.4. Call-Completion Is Invoked

The calling user indicates to the caller's agent that he wishes to invoke call-completion services on the recent call. Note that from the SIP point of view, the INVITE may be successful, but from the user's point of view, the call may be unsuccessful. E.g., the call may have connected to the callee's voicemail, which would return a

200 status to the INVITE but from the caller's point of view is "no reply".

The caller's agent subscribes to the call-completion event package[1] using the request URI of the original call. This SUBSCRIBE should be routed in much the same way as the original INVITE, but ultimately being routed not to the callee's UAs but to the callee's monitor. The Event header of the subscribe specifies the call-completion event package with a parameter "call_id=[Call-Id of the original call]".

The SUBSCRIBE should have headers to optimize its routing. In particular, it should contain "Request-Disposition: parallel, no-cancel".

The callee's monitor(s) that receive the SUBSCRIBE establish subscriptions. These subscriptions collectively represent the caller's agent's request for call-completion services. A callee's monitor MUST be prepared to receive multiple forks of a single SUBSCRIBE, and SHOULD respond 482 (Merged Request) to all but one fork. The callee's monitor MUST be prepared to receive SUBSCRIBEs regarding original calls that it has no knowledge of, and should respond 404 (Not Found) to such SUBSCRIBEs. The monitor may apply additional restrictions as to which caller's agents may subscribe.

The caller's agent MUST be prepared to receive multiple responses to the SUBSCRIBE and to have multiple subscriptions established. The agent MUST also be prepared to have the SUBSCRIBE fail, in which case, CC cannot be invoked for this original call.

The call-completion event package returns various information to the caller's agent, but the vital datum is that it contains an indication whether the callee's monitor has at this time chosen the caller's agent to perform the next CC call to the callee. This datum may initially be true, but more likely will initially be false and after a time progress to true.

3.5. The Call-Completion Request Is Queued

The continuation of the caller's agent's subscription indicates that the caller's agent is prepared to initiate the CC call when it is selected by the callee's monitor. If the caller's agent becomes unwilling to initiate the CC call (e.g., because the calling user has deactivated CC or because the caller's UA becomes busy), the caller's agent must terminate or suspend the subscription(s). (Currently, no method of suspending a subscription is defined.) If the caller's agent later becomes willing again to initiate CC for the original call, it may resume the suspended subscription(s) or initiate new one(s).

If the callee's monitor becomes aware that, according to its policy, the original call referenced by a subscription will never be selected for call-completion, it should terminate the subscription. (And reject any request to start a new subscription for that original call.)

3.6. Call-Completion Is Activated

The callee's monitor has a policy regarding when and how it selects CC requests to be activated. In addition to information regarding the callees' availability, this policy may take into account the circumstances of the original calls (which may distinguish CCNR vs. CCBS cases), the order in which the original calls arrived, and any previous CC attempts for the original calls. Usually the callee's monitor will choose only one CC request for activation at a time, but it may choose more than one.

The callee's monitor changes the "call completion active" datum for the chosen caller's agent from false to true. This triggers a notification for the agent's subscription.

The agent receives a notification for one of its subscriptions which contains the CC active datum set to true. It then terminates or suspends all other CC subscriptions for this original call, and all CC subscriptions for all other original calls, in order to prevent any other CC requests from this caller from being activated. The agent then determines whether the calling user is available for the CC call, possibly by calling the caller's UA(s) to see if it is answered.

If the calling user is not available, the caller's agent indicates this to the callee's monitor by terminating or suspending the activated CC subscription.

If the calling user is available, the caller's agent causes the caller's UA to initiate the CC call. The callee's monitor SHOULD provide a request URI to be used in the CC call as a datum in the call completion event package. If the callee's monitor does not provide a request URI, the caller's agent SHOULD use the request URI of the original call.

(Allowing the monitor to provide the request URI allows a number of advantages. The UA(s) supervised by the monitor probably correspond to an AOR that is more likely to route to those UA(s) than the original call's request URI (which the monitor has no control over). The URI can be a GRUU[5] in order to route the CC call to a specific UA. The URI can route to the monitor itself to allow the monitor to control completely the CC call's routing, including prioritizing it

over other calls. The URI can specify the original call's Call-Id, thus allowing correlation between it and the original call. The URI can contain authentication secrets to access special handling that is provided to CC calls.)

The callee's UA(s) and any associated proxies may give the CC call precedence over non-CC calls, depending on local policy.

The callee's monitor supervises the receiving of the CC call, possibly by the same mechanism that it supervised the receiving of the original call. If the caller's agent suspends or terminates its subscription, the monitor considers that agent to no longer be chosen for CC and takes further action according to its policy. If the CC call does not arrive at the callee's UA(s) promptly, the monitor may withdraw CC activation from the caller's agent by changing the value of its CC active datum to false. Similarly, if the CC call fails, the monitor may withdraw CC activation. Depending on its policy, the same original call may be selected again for CC activation at a later time. If the CC call succeeds, the monitor will also withdraw CC activation, and the original call will never again be selected for CC activation (and in practice, can be deleted from the monitor's records).

Question: Is that last statement true? Can a call appear to succeed from the monitor's point of view but fail from the calling user's point of view? Yes. Need a way to re-cycle CC.

Once the CC call has failed, or if it has succeeded, once the CC call has been terminated, the callee's monitor's policy may select another CC request for activation.

3.7. Data Provided in the Call-Completion Event Package

Question: What format should the event package data be presented in? [1] proposes a simple attribute-value format. We might also consider yet another XML format.

The only necessary information to be provided by the call-completion event package is the CC activation datum, whose value is false (meaning that this CC request has not been chosen for activation) or true (meaning that it has).

Question: If we decide to let the callee's monitor provide the request URI for the CC call, that request URI should probably be a mandatory datum as well.

The event package may provide information about the callee's monitor's policy. In particular, the PSTN CC feature gives an

indication of the "service retention" attribute, which indicates whether the CC request can be continued to a later time if the call-completion call fails due to the callee's UA(s) being busy.[1]

If the callee has a caller-queuing facility, we want to treat the call-completion queue as part of the queuing facility, and include in the event package information regarding the state of the queue, such as number of callers ahead of this caller and expected wait time. In that case, this data should probably not trigger a notification every time it changes, but rather at suitable time increments.

4. Security Considerations

The use of the CC facility allows the caller's agent to determine some status information regarding the callee. The information is confined to a available/not-available indication, and is to a considerable degree protected by the necessity of presenting the Call-Id of a recent call to the callee in order to obtain information.

The CC facility may enhance the effectiveness of SPIT by the following technique: The caller makes calls to a group of targets. The caller then requests CC for the calls that do not connect to the targets. The CC calls resulting are probably more likely to reach the targets than original calls to a further group of targets.

5. Revision History

5.1. Changes from draft-worley-call-completion-00 to draft-worley-call-completion-01

Correct author's contact information.

Various edits to improve clarity.

Based on Paul Kyzivat's critique, adjust description of when to make a CC call to use generic language about presence, rather than specific busy/non-busy state. Fundamentally, the ideal treatment of these states is not yet known, so we must allow quite a bit of latitude.

Clarify that the protocol does not distinguish CCNR from CCBS, because the implementation is the same, and the agents may not be able to tell the difference anyway.

Clarify that the agent is expected to establish multiple subscriptions with a single CC SUBSCRIBE.

Implement the concept that the monitor will supply a URI to be used for the CC call. Discuss the various applications of this scheme, as advanced by Paul Kyzivat and Scott Lawrence.

6. Normative References

[1] Poetzl, J., Huelsemann, M., and J. Stupka, "Extensions to the Session Initiation Protocol (SIP) for the support of the Call Completion Services for the European Telecommunications Standards Institute", I-D draft-poetzl-bliss-call-completion-00, December 2007.

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

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

[4] Sparks, R., "The Session Initiation Protocol (SIP) Refer Method", RFC 3515, April 2003.

[5] Rosenberg, J., "Obtaining and Using Globally Routable User Agent (UA) URIs (GRUU) in the Session Initiation Protocol (SIP)", draft-ietf-sip-gruu-11 (work in progress), October 2006.

Author's Address

Dale R. Worley
Bluesocket Inc.
10 North Ave.
Burlington, MA 01803
US

Phone: +1 781 229 0533 x173
Email: dworley@pingtel.com
URI: http://www.pingtel.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.