bliss
Internet-Draft
Intended status: Standards Track
Expires: August 15, 2013

D. Worley
Ariadne Internet Services, Inc.
M. Huelsemann
R. Jesske
Deutsche Telekom
D. Alexeitsev
TeleFLASH

February 11, 2013

Call Completion for Session Initiation Protocol (SIP) draft-ietf-bliss-call-completion-19

Abstract

The call completion feature defined in this specification allows the caller of a failed call to be notified when the callee becomes available to receive a call.

For the realization of a basic solution without queuing, this document references the usage of the dialog event package (RFC 4235) that is described as 'automatic redial' in the SIP Service Examples (RFC 5359).

For the realization of a more comprehensive solution with queuing, this document introduces an architecture for implementing these features in the Session Initiation Protocol where "call completion" implementations associated with the caller's and callee's endpoints cooperate to place the caller's request for call completion into a queue at the callee's endpoint, and when a caller's request is ready to be serviced, re-attempt of the original, failed call is made.

The architecture is designed to interoperate well with existing call-completion solutions in other networks.

material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 15, 2013.

Copyright Notice

Table of Contents

1. Introduction

The call completion (CC) feature allows the caller of a failed call to have the call completed without having to make a new call attempt while guessing when the callee becomes available. When the caller requests the use of the CC feature, the callee will be monitored for its availability. When the callee becomes available the callee will be given a certain timeframe for initiating a call. If the callee does not initiate a new call within this timeframe, then the caller will be recalled. When the caller accepts the CC recall then a CC call to the callee will automatically start. If several callers have requested the CC feature on the same callee, they will be recalled in a predefined order, which is usually the order in which they have requested the CC feature.

This draft defines the following CC features:

Call Completion on Busy Subscriber (CCBS): The callee is busy. The caller is recalled after the callee is not busy any longer.

Call Completion on No Reply (CCNR): The callee does not answer the call. The caller is recalled after the callee has completed a new call.

Call Completion on Not Logged-in (CCNL): The callee is not registered. The caller is recalled after the callee has registered again.

2. Requirements terminology

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 [RFC2119].

This document uses terms from [RFC3261].

3. Terminology

For the purpose of this service, we provide the following terminology:

Callee: a destination of the original call, and a target of the CC call.

Caller: The initiator of the original call and the CC request. The user on whose behalf the CC call is made.

Callee's monitor: a logical component which implements the call-completion queue for destination user(s)/UA(s), and performs the associated tasks, including sending CC recall events, analogous to the destination local exchange's role in SS7 CC.

Caller's agent: a logical component which makes CC requests and responds to CC recall events on behalf of originating user(s)/UA(s), analogous to the originating local exchange's role in SS7 CC.

CC, or call completion: a service which allows a caller who failed to reach a desired callee to be notified when the callee becomes available to receive a call.

CC activation: the indication by the caller to the caller's agent that the caller desires CC for a failed original call; this implies an indication transmitted from the caller's agent to the callee's monitor of the desire for CC processing.

CCBS, or Call Completion on Busy Subscriber: a CC service when the initial failure was that the destination UA was busy.

CCNR, or Call Completion on No Reply: a CC service when the initial failure was that the destination UA did not answer.

CCNL, or Call Completion on Not Logged-in: a CC service when the initial failure was that the destination UA was not registered.

CC call: a call from the caller to the callee, triggered by the CC service when it has determined that the callee is available.

CC indicator: an indication in the CC call INVITE used to prioritize the call at the destination.

CC possible indication: the data in responses to the INVITE of the original call which indicate that CC is available for the call.

CC recall: the action of the callee's monitor selecting a particular CC request for initiation of a CC call, resulting in an indication from the caller's agent to the caller that it is now possible to initiate a CC call.

CC recall events: event notifications of event package "call-completion", sent by the callee's monitor to the caller's agent to inform it of the status of its CC request.

CC recall timer: maximum time the callee's monitor will wait for the caller's response to a CC recall.

CC request: the entry in the callee's monitor queue representing the caller's request for CC processing, that is, the caller's call-completion subscription.

CC service duration timer: maximum time a CC request may remain active within the network.

CC queue: a buffer at the callee's monitor which stores incoming calls which are target for call completion. Note: This buffer may or may not be organized as a queue. The use of the term "queue" is by analogy with SS7 usage.

CCE, or call-completion entity: the representation of a CC request, or equivalently, an existing call-completion subscription within a callee's monitor's queue

Failed call: a call which does not reach a desired callee, from the caller's point of view. Note that a failed call may be successful from the SIP point of view; e.g., if the call reached the callee's voicemail, but the caller desired to speak to the callee in person, the INVITE receives a 200 response, but the caller considers the call to have failed.

Notifier: the user agent that generates NOTIFY requests for the purpose of notifying subscribers of the callee's availibility; for the CC service this is task of the callee's monitor.

Original call: the initial call which failed to reach a desired destination.

Retain option: a characteristic of the CC service; if supported, CC calls which again encounter a busy callee will not be queued again, but the position of the caller's entry in the queue is retained. Note that SIP CC always operates with the retain option active; a failed CC call does not cause the CC request to lose its position in the queue.

Subscriber: the user agent that receives NOTIFY requests with information of the callee's availibility; for the CC service this is task of the caller's agent.

Suspended CC request: a CC request which is temporarily not to be selected for CC recall.

4. Solution

4.1. Call-completion architecture

The call-completion architecture augments each caller's UA (or UAC) wishing to use the call-completion features with a "call-completion agent" (also written as "caller's agent").

It augments each callee's UA (or UAS) wishing to be the target of the call-completion features with a "call-completion monitor" (also written as "callee's monitor").

The caller's agent and callee's monitor functions can be integrated into the respective UAs, be independent end-systems, or be provided by centralized application servers. The two functions, though associated with the two UAs (caller and callee), also may be provided as services by the endpoints' home proxies or by other network elements. Though it is expected that a UA that implements call completion will have both functions so that it can participate in call completion as both caller and callee, the two functions are independent of each other.

A caller's agent may service more than one UA as a collective group if a caller or population of users will be shared between the UAs, and especially if the UAs share an AOR.

The caller's agent monitors calls made from the caller's UA(s) in order to determine their destinations and (potentially) their final response statuses, and the Call-Info header fields of provisional and final responses for to invoke the call completion feature.

A callee's monitor may service more than one UA as a collective group if a callee or population of users will be shared between the UAs, and especially if the UAs share an AOR. The callee's monitor may supply the callee's UAS(s) with Call-Info header field values for provisional and final responses.

The callee's monitor also instantiates a presence server used to monitor caller's availability for CC recall.

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

In order to allow flexibility and innovation, most of the interaction between the caller's agent, 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, the callee(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.

The caller's agent must be capable of performing a number of functions relative to the UA(s). The method by which it does so is outside the scope of this document, but an example method is described in Appendix A. The callee's monitor must be capable of performing a number of functions relative to the UA(s). The method by which it does so is outside the scope of this document, but an example method is described in Appendix B.

As a proof of concept, simple caller's agents and callee's monitors can be devised that interact with users and UAs entirely through standard SIP mechanisms [RFC6665], [RFC4235] and [RFC3515], as described in the Appendixes.

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 the callers determined to unsuccessful. The caller's agent monitors the status of the caller's UA(s) to determine when they are available to be used for a CC recall. The caller's agent can communicate to the caller's UA(s) that a CC recall is in progress and to inquire if the relevant caller is available for the CC recall.

The callee's monitor may utilize several methods to monitor the status of the callee's UA(s) and/or their users for availability to receive a CC call. This can be achieved through monitoring calls made to the callee's UA(s) to determine their caller's and their final response' statuses. And in a system with rich presence information, the presence information may directly provide this status. In a more restricted system, this determination can depend on the mode of the CC call in question, which is provided by the URI 'm' parameter. E.g., a UA is considered available for CCBS ("m=BS") when it is not busy, but a UA is considered available for CCNR ("m=NR") when it becomes not busy after being busy with an established call.

The callee's monitor maintains information about the set of INVITEs received by the callee's UA(s) considered unsuccessful by the caller. In practice, the callee's monitor may remove knowledge about an incoming dialog from its set if local policy at the callee's monitor establishes that the dialog is no longer eligible for CC activations.

4.2. Call-completion procedures

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. If the call-completion feature is available, the callee's monitor (note there can be a monitor for each of the callee's UAs) inserts a Call-Info header field with its URI and with "purpose=call-completion" in appropriate non-100 provisional or final response to the initial INVITE and

forwards them to the caller. The provisional response SHOULD be sent reliably, if the INVITE contained a Supported header field with the option tag 100rel. On receipt of a non-100 provisional or a final response with the indication that the call-completion feature is available, the calling user can invoke the CC feature.

The caller 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 have been successful, but from the user's point of view, the call may have been 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".

In order to receive information necessary for the caller to complete the call at the callee, the caller's agent subscribes to the call-completion event package at the callee's monitor.

The possibility of the caller to complete the call at the callee is also known as the call-completion state (cc-state) of the caller. The cc-states comprehend the values 'queued' and 'ready' (for call-completion).

In order to receive information from all destinations where the callee will be reachable, the caller's agent sends a SUBSCRIBE request for the call-completion event package to the original destination URI of the call and to all known callee's monitor URIs (which are provided by Call-Info header fields in provisional and final responses to the INVITE). The callee's monitor uses the subscription as an indication that caller is interested in using the CC feature with regard to the specified callee. The callee's monitor keeps a list or queue of the caller's agent's subscriptions, representing the requests from the caller's agent to the callee's monitors for call-completion services. These subscriptions are created, refreshed, and terminated according to the procedures of [RFC6665].

Upon receiving a SUBSCRIBE request from the caller's agent, the callee's monitor instantiates a presence state for the caller's UA which can be modified by the caller's UA to indicate its availability for CC call. The status at the presence upon instantiation is "open".

When the callee's monitor determines the callee and/or callee's UA available for a CC call, it selects a caller to execute the CC call and sends a call-completion event update (''cc-state: ready'') via a NOTIFY request to the selected caller's agent's subscription, telling it to begin the CC call to the callee's UA.

When the caller's agent receives this update, it initiates a CC recall by calling the caller's UA, and then starts the CC call to the callee's UA, using 3rd party call control procedures in accordance with [RFC3725]. The caller's agent can also check by other means whether the caller is available to initiate the CC call to the callee's UA. If the caller is available, the caller's agent directs the caller's UA to initiate the CC call to the callee's UA.

The caller's agent marks the CC call as such by adding a specific SIP URI parameter to the Request-URI, so it can be given precedence by the callee's monitor in reaching the callee's UA.

If the caller is not available on the receipt of the "ready for recall" notification, the caller's agent suspends the CC request at the callee's monitor by sending a PUBLISH request containing presence information to the callee's monitor's presence server, informing about the presence status 'closed'. Once the caller becomes available for a CC call again, the caller's agent resumes the CC request by sending another PUBLISH request to the callee's monitor, informing about the presence status 'open'.

On the receipt of the suspension request, the callee's monitor performs the monitoring for the next non-suspended CC request in the queue. On the receipt of the resume from the previously suspended caller's agent that was at the top of the queue, the callee's monitor performs the callee monitoring for this caller's agent.

When the CC call fails there are two possible options: the CC feature has to be activated again by caller's agent subscribing to callee's monitor, or CC remains activated and the original CC request retains its position in the queue if retain option is supported.

The retain option (see section 3) determines the callee's monitor's behavior when a CC call fails. If the retain option is supported, CC remains activated and the original CC request retains its position in the queue. Otherwise the CC feature is deactivated, and the caller's agent would have to subscribe again to reactivate it.

A monitor that supports the retain option provides the cc-service-retention header in its call-completion events. A caller's agent that also supports the retain option uses the presence of this header to know not to generate a new CC request after a failed CC call.

Monitors not supporting the retain option do not provide the cc-service-retention header. A failed CC call causes the CC request to be deleted from the queue, and these monitors will terminate the corresponding caller's agent's subscription to inform that agent that its CC request is no longer in the queue. A caller's agent that does

not support the retain option can also terminate its subscription when a CC call fails, so it is possible that both the caller's agent and the monitor may be signalling the termination of the subscription concurrently. This is a normal SIP Events [[RFC6665]] scenario. After the subscription is terminated, the caller's agent may create a new subscription (as described in section 6.2) to reactivate the CC feature for the original call.

4.3. Automatic redial as a fallback

Automatic redial is a simple end-to-end design. An automatic redial scenario is described in [RFC5359], section 2.17. This solution is based on the usage of the dialog event package. When the callee is busy when the call arrives, the caller subscribes to the callee's call state. The callee's UA sends a notification when the callee's call state changes. This means the caller is also notified when the callee's call state changes to 'terminated'. The caller is alerted, then the caller's UA starts a call establishment to the callee again. If several callers have subscribed to a busy callee's call state, they will be notified at the same time that the call state has changed to 'terminated'. The problem of this solution is, that it might happen that several recalls are started at the same time. This means it is a heuristic approach with no guarantee of success.

There is no interaction between call completion and automatic redial, as there is a difference in the behavior of the callee's monitor and the caller when using the dialog event package for receiving dialog information or for aggregating a call completion state.

4.4. Differences from SS7

SIP call completion differs in some ways from the CCBS and CCNR features of SS7 (which is used in the PSTN). For ease of understanding, we enumerate some of the differences here.

As there is no equivalent to the forking mechnism in SS7, in the PSTN calls can be clearly differentiated between successful or unsuccessful. Due to the complex forking situations that are possible in SIP, a call may "fail" from the point of view of the user and yet have a "success" response from SIP's point of view. (This can happen even in simple situations: e.g., a call to a busy user that fails over to his voicemail receives a SIP success response, even though the caller may consider it "busy subscriber".) Thus, the caller must be able to invoke call completion even when the original call appeared to succeed. To support this, the caller's agent must record successful calls as well as unsuccessful calls.

In SIP, only the caller's UA or service system on the originating

side and the callee's UA or service system on the terminating side need to support call completion for call completion to work successfully between the UAs. Intermediate SIP systems (proxies or B2BUAs) do not need to implement call completion; they only need to be transparent to the usual range of SIP messages. In the PSTN also intermediate nodes like media gateway controllers have to implement the call completion service.

5. Call-completion queue model

The callee's monitor manages CC for a single URI. This URI is likely to be a published AOR, or more likely "non-voicemail AOR", but it may be as narrowly scoped as a single UA's contact URI. The callee's monitor manages a dynamic set of call-completion entities (called "CCEs") which represent CC requests, or equivalently, the existing incoming call-completion subscriptions. This set is also called a queue, because a queue data structure often aids in implementing the callee's monitor's policies for selecting CCEs for CC recall.

Each CCE has an availability state, determined through caller's presence status at the callee's monitor. A presence status of ''open'' represents CCE's availability state of 'available' and a presence status of "closed" represents CCE's availability state of 'unavailable'.

Each CCE has a recall state which is visible via subscriptions. The recall state is either "queued" or "ready".

Each CCE carries the From URI of the SUBSCRIBE request that caused its creation.

CC subscriptions arrive at the callee's monitor by addressing the URIs the callee's monitor returns in Call-Info header fields. The request URI of the SUBSCRIBE request determines the queue to which the resulting CCE is added. The resulting subscription reports the status of the queue. The base event data is the status of all the CCEs in the queue, but the data returned by each subscription is filtered to report only the status of that subscription's CCE. (Further standardization may define means for obtaining more comprehensive information about a queue.)

When a CCE is created, it is given the availability state "available" and recall state "queued".

When the callee's monitor receives PIDF bodies [RFC3863] via PUBLISH requests [[RFC3903]], these PUBLISH requests are expected to be sent by subscribers to indirectly suspend and resume their CC requests by

modifying its CCE availability state. A CCE is identified by the request-URI (if it was taken from a call-completion event notification which identifies the CCE) or the From URI of the request (matching the From URI recorded in the CCE). Receipt of a PUBLISH with 'status' of 'open' sets the availability state of the CCE to 'available' (resume); 'status' of 'closed' sets the availability state of the CCE to 'not-available' (suspend).

A CC request is eligible for recall only when its CCE's availability state is "available" and the "m" value of the CCE also indicates an available state. The callee's monitor MUST NOT select for recall any CC requests that fail to meet those criteria. Within that constraint, the callee's monitor's selections are determined by its local policy. Often, a callee's monitor will choose the acceptable CCE that has been in the queue the longest. When the callee's monitor has selected a CCE for recall, it changes the CCE's recall state from 'queued' to 'ready', which triggers a notification on the CCE's subscription.

If a selected subscriber then suspends its request by sending a PUBLISH with the presence status 'closed', the CCE becomes not-available, and the callee's monitor changes the CCE's recall state to 'queued'. This may cause another CCE (e.g., that has been in the queue for less time) to be selected for recall.

The caller's presence status at the callee's monitor is terminated when the caller completes its CC call or when the caller's agent's subscription at the callee's monitor is terminated.

6. Caller's agent behavior

6.1. Receiving the CC possible indication

The caller's agent MUST record the From URI and SHOULD record the final request status that the caller's UA received along with the contents of Call-Info header fields of provisional and final responses.

Note that receiving a CC possible indication also depends on the aggregation of final responses by proxies, in case of 4xx responses some 4xx responses are more likely to be sent to the caller.

6.2. Subscribing to CC

For CC activation the caller's agent MUST send a SUBSCRIBE to all known callee's monitor URIs. A callee's monitor URI may be provided in the Call-Info header field in provisional and final responses to

the INVITE sent back by the callee's monitor(s). Additionally, the caller's agent SHOULD include the original request-URI that it sent the original INVITE to, in its set of callee's monitor URIs, when it is unclear if the call has forked to additional callees whose responses the caller has not seen. A SUBSCRIBE to the original request-URI alone is used in cases where the caller's agent has not received or does not remember any callee's monitor URI. The caller's agent SHOULD add an 'm' parameter to these URIs in order to indicate the desired call-completion proceeding at the callee's monitor. The 'm' parameter SHOULD have the value of the 'm' parameter received in the Call-Info header field of the responses to the original INVITE.

To minimize redundant subscriptions, these SUBSCRIBEs SHOULD be presented as forks of the same transaction as defined by section 8.2.2.2 of [RFC3261], if the caller's agent is capable of doing so.

The agent MUST NOT maintain more than one CC request for a single caller and directed to a single original destination URI. If a caller requests CC a second time for the same destination URI, the agent MUST consolidate the new request with the existing CC request by either reusing the existing CC subscriptions or terminating and then recreating them. For this purpose, equality of callers is determined by comparing caller's AORs and equality of destination URIs is determined by comparing them per [RFC3261] section 19.1.4.

When generating these SUBSCRIBEs, the From URI MUST be the caller's AOR. The To URI SHOULD be the destination URI of the original call (if the agent knows that and can insert it into the To header), and otherwise MUST be the request-URI of the SUBSCRIBE.

The SUBSCRIBE SHOULD have header fields to optimize its routing. In particular, it SHOULD contain "Request-Disposition: parallel", and an Accept-Contact header field to eliminate callee UAs that are not acceptable to the caller.

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

If the caller's agent no longer wants to initiate the CC call (e.g., because the caller has deactivated CC), the caller's agent terminates the subscription in accordance with [RFC6665] or suspends the subscription(s) as specified in subclause 6.5.

6.3. Receiving a CC recall notification

When receiving a NOTIFY with the cc-state set to 'ready', the caller's agent SHOULD suspend all other subscriptions to CC, by following the step in section 6.5, in order to prevent any other CC requests from this caller to receive CC recalls. The caller's agent starts the CC recall to the caller by confirming that the caller would be able to initiate a CC call, e.g. by calling the caller's UA(s).

6.4. Initiating a CC call

If the caller is available for the CC call and willing to initiate the CC call, the caller's agent causes the caller's UA to generate a new INVITE towards the callee. The caller's UA MAY add a 'm' URI parameter with the value of the 'm' parameter received in Call-Info header in the response to original INVITE, in order to specify his preferences in CC processing and to prioritize the CC call. The INVITE SHOULD be addressed to the URI specified in the cc-URI of the NOTIFY, or if not available it SHOULD use the URI in the Call-Info header field of the response to the original INVITE, or if not available it MAY use the request-URI of the original INVITE, if this URI was recorded. Note that the latter choice may not provide ideal routing, but in simple cases it is likely to reach the desired callee/callee's monitor.

6.5. Suspending CC

If the caller is not available for the CC recall, the CC request SHALL be suspended by the caller's agent until the caller becomes available again, or if the conditions relevant to the caller's agent's local policy for suspensions have changed. To suspend the CC request, the caller's agent SHALL publish the caller's presence state by sending a PUBLISH request to each callee's monitor where the presence server for CC resides in accordance with the procedures described in [RFC3903] , giving the PIDF state 'closed' for the caller's identity as presentity. The PUBLISH request SHOULD contain an Expires header field with a value that corresponds to the current value of the remaining CC subscription duration.

Each PUBLISH SHOULD be sent to the CC URI as received in the NOTIFY, or within the corresponding SUBSCRIBE dialog, or if that is not possible, to the corresponding callee's monitor URI received in the Call-Info header field of the NOTIFY, or if one is not available, the Contact address of the subscription.

6.6. Resuming CC

When the caller is no longer busy, or if the conditions relevant to the caller's agent's suspension policy have changed, then the CC request SHALL be resumed by the caller's agent. To resume a CC request, the caller's agent SHALL publish the Caller's presence state by sending a PUBLISH request to each callee's monitor a PUBLISH request in accordance with the procedures described in [RFC3903] , informing about the PIDF state 'open' but otherwise be constructed as same as the suspend PUBLISH request. These PUBLISH requests are sent to presence server that are instantiated at a CC monitor.

In the case where the caller's agent has sent several CC suspension requests to different callee's monitors and the caller becomes available again, as determined by the caller's agent's local policy about resumption the caller's agent MAY send a PUBLISH to resume a CC request to each callee's monitor for which there is a suspended CC request. Note that the caller's agent's policy about resumption may prescribe a manual resumption and thus a suspended CC request should not be automatically resumed.

7. Callee's monitor behavior

7.1. Sending the CC possible indication

The callee's monitor MUST record the From URI and MAY record the final request status(es) returned by the callee's UA(s).

If the callee's monitor wants to enable the caller to make use of the CC service, it MUST insert a Call-Info header field with "purpose=call-completion" in the final response message (e.g. in a 486 to enable call-completion due to busy subscriber) and at least one non-100 provisional response message (e.g. in a 180 due to no response) to the initial INVITE when forwarding it to the caller. The non-100 provisional response message SHOULD be sent reliably if the INVITE contained a Supported header field with the option tag 100rel. The Call-Info header field values defined in this specification positively indicates that CC is available for the failed fork of the call.

The callee's monitor SHOULD insert a URI in the Call-Info header field where the caller's agent should subscribe for call-completion. Ideally, it is a globally-routable URI [RFC5627] for the callee's monitor. In practice, it may be the callee's AOR, and the SUBSCRIBE will be routed to the callee's monitor only because it specifies "Event: call-completion".

In order to enable call-completion, the Call-Info header field MUST be set up according to the following scheme:

Call-Info:monitor-URI;purpose=call-completion;m=XX

The 'm' parameter defines the "mode" of call completion. The "m=NR" parameter indicates that it failed due to lack of response, the "m=BS" parameter indicates that it failed due to busy subscriber, and the "m=NL" parameter indicates that it failed due to non registered subscriber (no devices are registered for the AoR contacted). The 'm' parameter is useful for PSTN interworking and assessing presence information in the callee's monitor. It is possible that other values will be defined in future. It is also allowed to omit the 'm' parameter entirely. Implementations MUST accept CC operations in which the 'm' parameter is missing or has an unknown value, and execute them at its best in their environment (which is likely to be a degraded service, especially when interoperating with SS7).

7.2. Receiving a CC subscription

The callee's monitor MUST be prepared to receive SUBSCRIBEs for the call-completion event package directed to the URIs of UA(s) that it is servicing and any URIs that the callee's monitor provides in Call-Info header fields. The SUBSCRIBEs MUST be processed in accordance with the procedures defined in [RFC6665].

The callee's monitor(s) that receive the SUBSCRIBE establish subscriptions. These subscriptions represent the caller's agent's request for call-completion services.

If the monitor receives two or more SUBSCRIBEs that have the same Call-Id header field value and the monitor considers the request-URIs of the received SUBSCRIBEs to request the status of the same set of UAs, then they are redundant forks of one SUBSCRIBE request, and the monitor SHOULD reject all but one of the requests with 482 (Merged Request) responses.

The monitor MAY determine that an incoming CC SUBSCRIBE is a duplicate of an existing CC subscription if: (1) the Call-Id header field values are different, (2) the From URIs (i.e., the caller's AORs) are the same (per [RFC3261] section 19.1.4), (3) the To URIs (which should be the request-URI of the original call) have the same user and hostpart components, and (4) the monitor considers the request-URIs of the received SUBSCRIBEs to request the status of the same set of UAs.

If the monitor determines that a new subscription is a duplicate of an existing subscription, it MAY terminate the existing subscription

in accordance with the procedures defined in [RFC6665]. In any case, it MUST establish the new subscription.

The callee's monitor may apply restrictions as to which caller's agents may subscribe.

The continuation of the caller's agent's subscription indicates to the callee's monitor that the caller's agent is prepared to initiate the CC call if it is selected for the 'ready' state. If the callee's monitor becomes aware of a subscription which cannot be selected for a CC recall, it SHOULD terminate the subscription in accordance with [RFC6665].

7.3. Sending a CC notification

The call-completion event package returns various information to the caller's agent, but the vital datum it contains is the cc-state of the caller's agent's CC request in the CC queue, which in the beginning is 'queued'. When the cc-state of the agent's request changes, the callee's monitor MUST send a NOTIFY for a call-completion event to the caller's agent. The notification SHOULD also contain a URI which can be used for suspension requests. Ideally, it is a globally-routable URI [RFC5627] for the callee's monitor. In practice, it may be the callee's AOR, and the SUBSCRIBE will be routed to the callee's monitor only because it specifies "Event: call-completion".

The call-completion event package provides limited information about the callee's monitor's policy. In particular, like in the PSTN, the "'cc-service-retention" datum gives an indication of the "service retention" attribute, which indicates whether the CC request can be continued to a later time if the CC call fails due to the callee's UA(s) being busy. If the callee's monitor supports the service-retention option, the callee's monitor SHOULD include the cc-service-retention parameter.

The callee's monitor has a policy regarding when and how it selects CC requests for the recall. This policy may take into account the type of the requests (e. g. CCNR vs. CCBS), the state of the callee's UA(s), the order in which the CC requests arrived, the length of time the CC requests have been active, and any previous attempts of CC activations for the same original call. Usually the callee's monitor will choose only one CC request for the recall at a time, but if the callee's UA(s) can support multiple calls, it may choose more than one. Usually the callee's monitor will choose the oldest active request.

When the callee's monitor changes the state datum for the chosen

subscription from "queued" to "ready", the callee's monitor MUST send a NOTIFY for the caller's agent's subscription with the cc-state set to 'ready' (recall notification). The NOTIFY SHOULD also contain in the cc-URI a URI to be used in the CC call. In practice, this may be the AOR of the callee.

Upon sending the recall notification the callee's monitor MUST start a recall timer. It is RECOMMENDED to use a value between 10 and 20 seconds, which corresponds to the recommendation for the call completion services in ETSI [ETS300.356-18] and ITU-T [ITU-T.Q.733].

7.4. Receiving a CC call

The callee's UA(s) and the callee's monitor may give the CC call precedence over non-CC calls by evaluating the presence of the 'm' URI parameter and the From header of the INVITE request. The callee's monitor supervises the receiving of the CC call. Upon arrival of the CC call the recall timer MUST be stopped. If the CC call does not arrive at the callee's UA(s) before the expiry of the recall timer, the callee's monitor SHOULD stop processing the recall and change the value of the cc-state datum to "queued" if it supports the retain option and terminate the subscription along with the queue if it doesn't support the retain option. Similarly, if the CC call is not accepted, the callee's monitor will stop the CC recall processing. Depending on its policy, the same original call may be selected again for a CC recall at a later time. If the CC call succeeds, the callee's monitor MUST terminate the relevant subscription in accordance with [RFC6665], and MUST remove any associated presence event state used for suspend and resume for the caller of the CC call.

Once the CC call has been terminated, successfully or unsuccessfully, the callee's monitor's policy MAY select another CC request for a recall according to the callee's monitor's policy. Note that according to the callee's monitor's policy several recalls may be processed at the same time.

7.5. Receiving a CC suspension

The monitor may receive PUBLISH requests to suspend CC requests from caller's agent as described in section 6.5. The PUBLISH requests may be received via the URI it manages, any URI that it inserts into a Call-Info header, any contact URI it uses as a notifier for "call-completion" events, or any URI it returns as the "URI" line of the call-completion event packages.

The receipt of the PUBLISH request initiates a presence event state for the caller's identity at the presence server functionality of the

callee's monitor as specified in [RFC3903] , together with a logical presence server if this has not been done before for another call.

Note: The presence server may initiate a presence event state for the caller's identity at the receipt of SUBSCRIBE request as well, dependent on the implementation.

The monitor SHOULD identify the addressed CCE by the request-URI of the PUBLISH request, or if that is not possible, by the From URI.

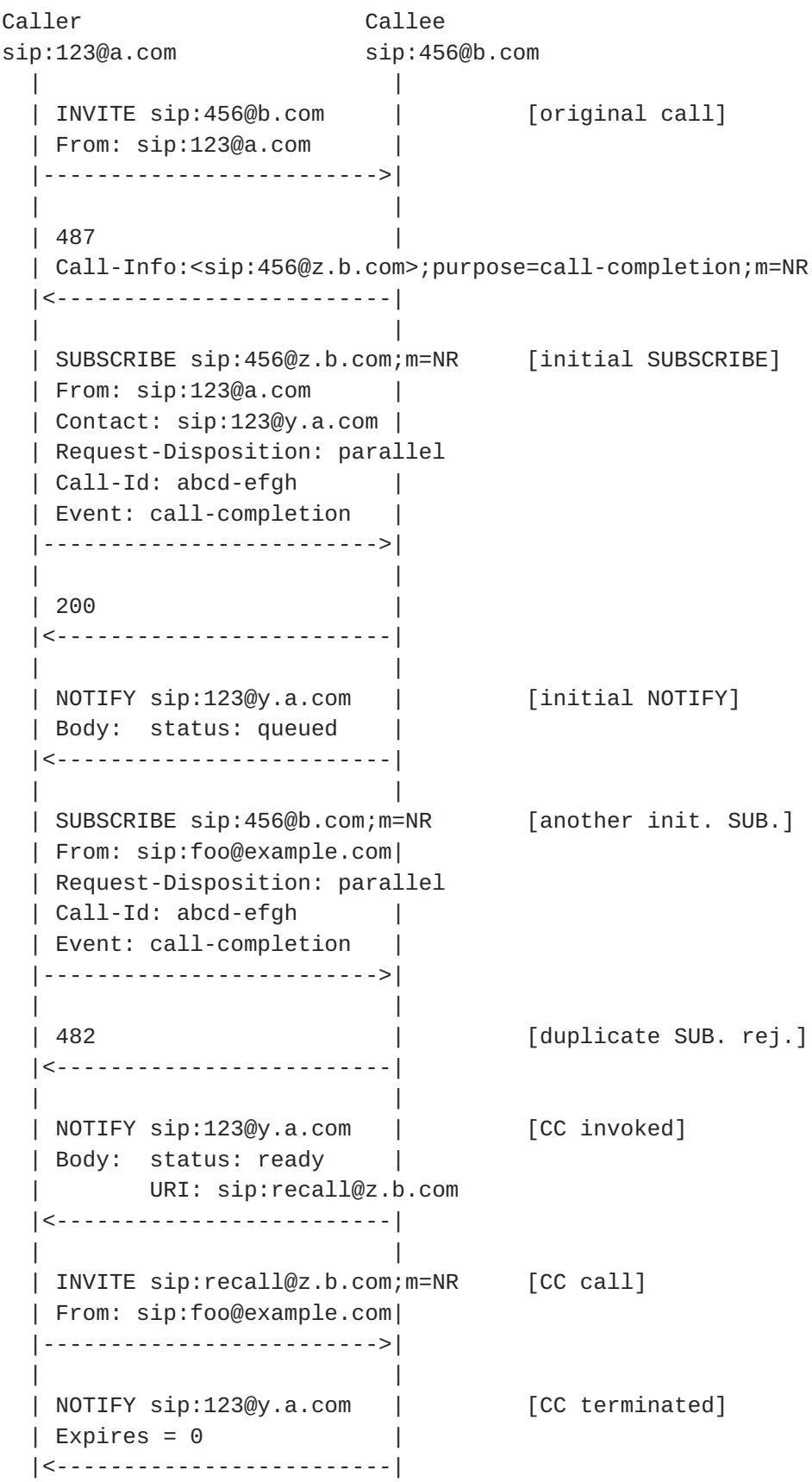
If the processing of a CC request results in suspending that CC request by receiving a PUBLISH request from caller's agent as described in section 6.5, the callee's monitor MUST stop the recall timer and MUST ensure that the request is set to a 'queued' state, and then the callee's monitor MUST attempt to process another CC request in the queue according to the callee's monitor's local policy.

7.6. Receiving a CC resumption

When a CC request becomes resumed by receiving a PUBLISH request from caller's agent as described in section 6.6, the presence event state for the caller's identity at the presence server functionality of the CC monitor MUST be modified as described in [RFC3903]. If the callee is not busy and there is no entry in the CC queue which is currently being processed, the callee's monitor MUST process the queue as described in section 7.3 above.

8. Examples

A basic call flow, with only the most significant messages of a call-completion activation and invocation shown, is as follows (please note this is an example and there may be variations in the failure responses):


Caller
sip:123@a.com
Callee
sip:456@b.com
INVITE sip:456@b.com
From: sip:123@a.com
[original call]
487
Call-Info:<sip:456@z.b.com>;purpose=call-completion;m=NR
SUBSCRIBE sip:456@z.b.com;m=NR
From: sip:123@a.com
Contact: sip:123@y.a.com
Request-Disposition: parallel
Call-Id: abcd-efgh
Event: call-completion
[initial SUBSCRIBE]
200
NOTIFY sip:123@y.a.com
Body: status: queued
[initial NOTIFY]
SUBSCRIBE sip:456@b.com;m=NR
From: sip:foo@example.com
Request-Disposition: parallel
Call-Id: abcd-efgh
Event: call-completion
[another init. SUB.]
482
[duplicate SUB. rej.]
NOTIFY sip:123@y.a.com
Body: status: ready
URI: sip:recall@z.b.com
[CC invoked]
INVITE sip:recall@z.b.com;m=NR
From: sip:foo@example.com
[CC call]
NOTIFY sip:123@y.a.com
Expires = 0
[CC terminated]

The original call is an ordinary INVITE. It fails due to no-response (ring-no-answer). In this case, the callee's governing proxy generates a 487 response because the proxy canceled the INVITE to the UA when it rang too long without an answer. The 487 response carries a Call-Info header field with "purpose=call-completion". The Call-Info header field positively indicates that CC is available for this failed fork of the call. The "m=NR" parameter indicates that it failed due to no-response, which is useful for PSTN interworking and assessing presence information in the callee's monitor.

The URI in the Call-Info header field (<sip:456@z.b.com>) is where the caller's agent should subscribe for call-completion processing. Ideally, it is a globally-routable URI for the callee's monitor. In practice, it may be the callee's AOR, and the SUBSCRIBE will be routed to the callee's monitor only because it specifies "Event: call-completion".

CC is activated by sending a SUBSCRIBE to all known callee's monitor URIs. These can be provided by the Call-Info header field in the response to the INVITE.

Additionally, the caller's agent needs to include the original request-URI in its set of callee's monitor URIs, because the call may have forked to additional callees whose responses the caller has not seen. (A SUBSCRIBE to the request-URI alone is used in cases where the caller's agent has not received or cannot remember any callee's monitor URI.)

The caller's agent adds to these URIs an 'm' parameter (if possible). In this case, the caller's agent forks the SUBSCRIBE to two destinations as defined by section 8.2.2.2 of [RFC3261], with appropriate Request-Disposition. The first SUBSCRIBE is to the URI from Call-Info.

The second SUBSCRIBE is to the original request-URI, and reaches the same callee's monitor. Because it has the same Call-Id as the SUBSCRIBE that has already reached the callee's monitor, the callee's monitor rejects it with a 482, thus avoiding redundant subscriptions.

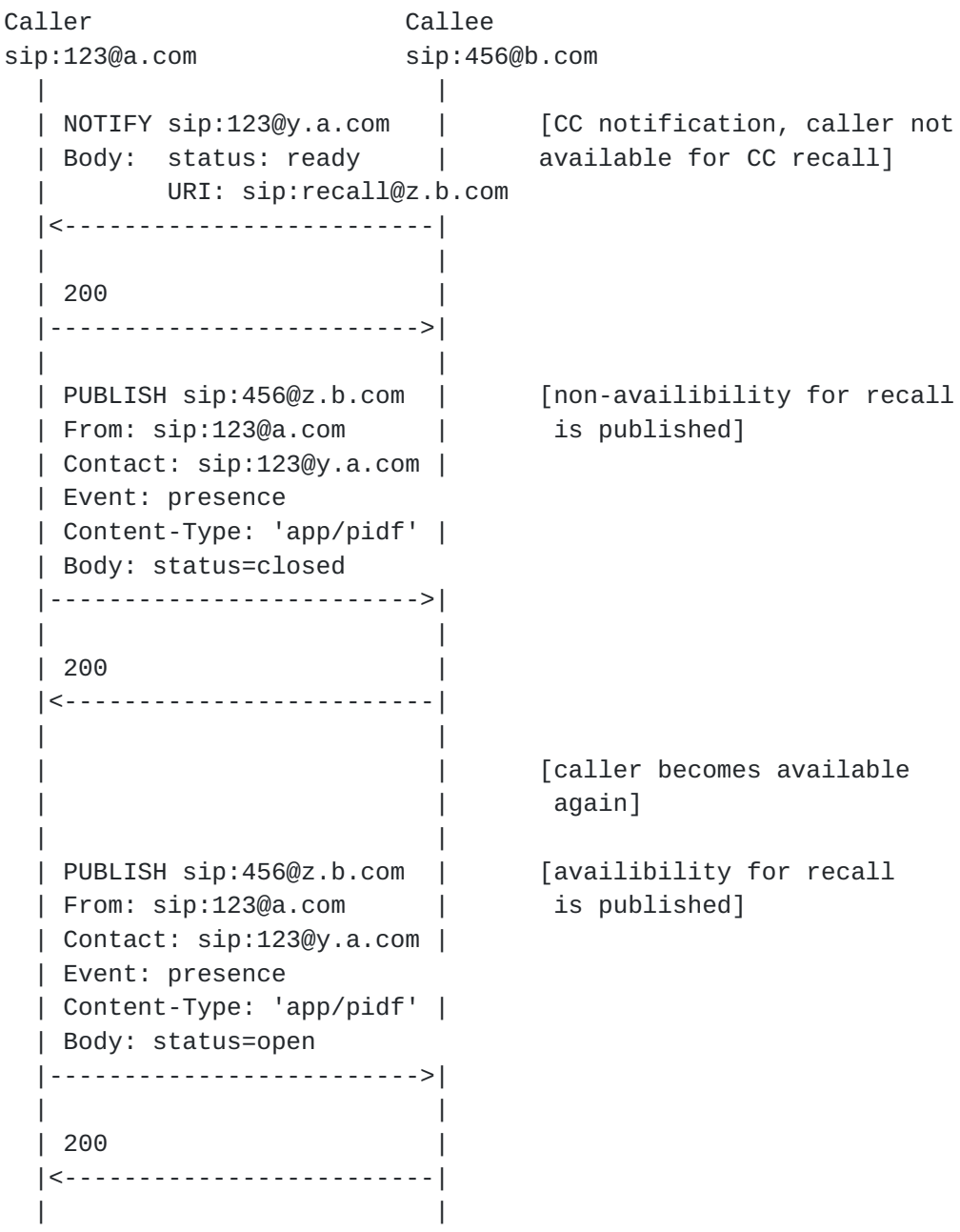
The initial NOTIFY for the successful SUBSCRIBE has "state: queued" in its body. Eventually, this caller is selected for CC, and is informed of this via a NOTIFY containing "state: ready". This NOTIFY carries a URI to which the INVITE for CC call should be sent. In practice, this may be the AOR of the callee.

The caller generates a new INVITE to the URI specified in the NOTIFY, or if there was no such URI or if the caller's agent cannot remember it, it may use the original request-URI. The caller adds the 'm'

parameters (if possible), to specify CC processing.

Finally the subscription for the CC request is terminated by the callee's monitor.

Another flow, with only the most significant messages of call-completion suspension and resumption shown, is as follows:



The caller is selected for CC, and is informed of this via a NOTIFY request containing "state: ready". At this time, the caller is not available for the CC recall.

For updating his presence event state at the presence server functionality at the callee, the caller generates a PUBLISH request to the CC URI as received in the NOTIFY, or within the corresponding SUBSCRIBE dialog, or if that is not possible, to the corresponding callee's monitor URI received in the Call-Info header field of the NOTIFY, or if one is not available, the Contact address of the subscription, informing about the PIDF state 'closed' .

When the caller is again available for the CC recall, he caller updates his presence event state at the presence server functionality at the callee by generating a PUBLISH request informing about the PIDF state 'open', but otherwise constructed as same as the suspend PUBLISH request.

9. Call-completion event package

This section specifies the call-completion event package, in accordance with section 4.4 of [RFC6665]. The call-completion event package has the media type "application/call-completion".

Note that if the callee has a caller-queuing facility, the callee's monitor may 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. How this information is conveyed is left for further standardization.

9.1. Event package name

The SIP Events specification requires package definitions to specify the name of their package or template-package. The name of this package is "call-completion". This value appears in the Event and Allow-events header fields.

9.2. Event package parameters

No package-specific Event header field parameters are defined for this event package.

9.3. SUBSCRIBE bodies

[RFC6665] requires package definitions to define the usage, if any, of bodies in SUBSCRIBE requests.

The SUBSCRIBE request MAY contain an Accept header field. If no such header field is present, it has a default value of "application/call-completion". If the header field is present, it MUST include "application/call-completion".

A SUBSCRIBE request for a call-completion package MAY contain a body. This body defines a filter to be applied to the subscription. Filter documents are not specified in this document, and may be the subject of future standardization activity.

A SUBSCRIBE request requests call-completion information regarding calls recently made from the same caller to the callee UA(s) serviced

by the notifier. Calls are defined to be "from the same caller" if the URI-part of the From header field value in the INVITE is the same as the URI-part of the From header field value in the SUBSCRIBE.

9.4. Subscribe duration

[RFC6665] requires package definitions to define a default value for subscription durations, and to discuss reasonable choices for durations when they are explicitly specified.

If a SUBSCRIBE does not explicitly request a duration, the default requested duration is 3600 seconds, as that is the highest service duration timer value recommended for the call completion services in ETSI [ETS300.356-18] and ITU-T [ITU-T.Q.733]. As because of the subscription duration no explicit timer is needed, and the subscription duration can be seen as an equivalent to the SS7 service duration timer, this specification refers to the subscription duration also as the service duration timer. It is RECOMMENDED that subscribers request, and that notifiers grant, a subscription time of at least 3600 seconds.

If a notifier can determine that, according to its policy, after a certain duration the requested subscription can not any more proceed to "ready" state, it SHOULD reduce the granted subscription time to that duration. If a notifier can determine that, according to its policy, the requested subscription can never proceed to "ready" state, it should refuse the subscription.

9.5. NOTIFY bodies

[RFC6665] requires package definitions to describe the allowed set of body types in NOTIFY requests, and to specify the default value to be used when there is no Accept header field in the SUBSCRIBE request. A NOTIFY for a call-completion package MUST contain a body that describes the call-completion states.

As described in [RFC6665], the NOTIFY message will contain bodies that describe the state of the subscribed resource. This body is in a format listed in the Accept header field of the SUBSCRIBE, or in a package-specific default format if the Accept header field was omitted from the SUBSCRIBE.

In this event package, the body of the notification contains a call-completion document. All subscribers and notifiers MUST support the "application/call-completion" data format described in section 10. The SUBSCRIBE request MAY contain an Accept header field. If no such header field is present, it has a default value of "application/call-completion". If the header field is present, it MUST include

"application/call-completion". Of course, the notifications generated by the server MUST be in one of the formats specified in the Accept header field in the SUBSCRIBE request.

9.6. Subscriber generation of SUBSCRIBE requests

Subscribers MUST generate SUBSCRIBE requests when they want to subscribe to the call-completion event package at the terminating side in order to receive call-completion notifications. The generation of SUBSCRIBE requests can imply the usage of a call-completion service specific timer as described in section 9.4.

9.7. Notifier processing of SUBSCRIBE requests

Upon receiving a subscription refresh, the notifier MUST set the "expires" parameter of the Subscription-State header field to a value not higher than the current remaining duration of the subscription regardless of the value received in the Expires header field (if present) of the subscription refresh.

If a subscription is not successful because the call-completion queue has reached the maximum allowed number of entries (short term denial), the notifier MUST send a 480 Temporarily Unavailable response to the subscriber, possibly with a Retry-after header field in accordance with the notifier's policy. If a subscription is not successful because a condition has occurred that prevents and will continue to prevent the call-completion service (long term denial), the notifier MUST send a 403 Forbidden response to the subscriber.

A notifier MAY receive multiple forks of the same SUBSCRIBE, as defined by section 8.2.2.2 of [RFC3261]. In such a case, the notifier MUST reject all but one of the SUBSCRIBEs with a 482 Merged Request response unless some other failure response applies.

The call-completion information can be sensitive. Therefore, all subscriptions SHOULD be handled with consideration of the security considerations discussed in section 11, in particular for verifying the identity of the subscriber.

9.8. Notifier generation of NOTIFY requests

Notifiers MUST generate NOTIFY requests when the CC request's state changes to 'queued' or to 'ready (for call-completion)'. A NOTIFY that is sent with non-zero expiration MUST contain the "cc-state" parameter. The parameter's value MUST be "queued" if the call-completion request represented by the subscription is not at this time selected by the callee's monitor for CC recall, and the parameter's value MUST be "ready" if the request is at this time

selected by the callee's monitor for CC recall.

A NOTIFY sent with a zero expiration (e.g., as a confirmation of a request to unsubscribe) MAY contain the "cc-state" parameter.

When the callee's monitor withdraws the selection of the request for the CC recall (e.g., because the caller's agent has not initiated the CC recall INVITE before the CC recall timer expires, or because the agent has suspended the request from being considered for CC recall), the notifier MUST send a NOTIFY to the subscription of the selected request. This NOTIFY MUST contain the "cc-state" parameter set to "queued".

If the call-completion subscription was successful and the retain option is supported at the callee, the NOTIFY MUST contain the "cc-service-retention" parameter.

9.9. Subscriber processing of NOTIFY requests

When receiving a NOTIFY requests with the cc-state set to 'ready', subscribers SHOULD suspend all other CC subscriptions for the original call at other notifiers. The receipt of a NOTIFY request with the cc-state set to 'ready' by the subscriber will also cause an interaction with the instances at the subscribers side hat are responsible for starting the CC recall.

9.10. Handling of forked requests

Forked requests are expected to be common for the call-completion event type. The subscriber MUST be prepared to process NOTIFY requests from multiple notifiers and to coordinate its processing of the information obtained from them in accordance with the procedures in this document.

9.11. Rate of notifications

The call completion service typically involves a single notification per notifier and per subscription that notifies about the change to 'ready (for call-completion)', but MAY involve several notifications about the change to the 'ready' state, separated by a call completion call that failed due to a busy callee . Typically, notifications will be separated by at least tens of seconds. Notifiers SHOULD NOT generate more than three notifications for one subscription in any ten-second interval. Since it is important to avoid useless recalls, a notifier SHOULD send state changes to "queued" from "ready" promptly. Thus, a notifier SHOULD NOT send a state change to "ready" as the third notification in a ten-second interval, as that would make it impossible to promptly send a further state change to

"queued".

9.12. State agents

State agents have no defined role in the handling of the call-completion package.

10. Call-completion information format

The following syntax specification uses the Augmented Backus-Naur Form (ABNF) as described in [RFC5234]. The formal syntax for the application/call-completion MIME type is described below. In general, the call-completion body is to be interpreted in the same way as SIP headers: (1) the names of the lines are case-insensitive, (2) the lines can be continued over line boundaries if the succeeding lines start with horizontal white space, and (3) lines with unknown names are to be ignored. The header lines defined in this document can occur at most once in any given call-completion document.

```
call-completion = 1*(cc-header CRLF)

cc-header = cc-state / cc-service-retention / cc-URI / extension-
header
```

The above rules whose names start with "cc-" are described below. Other rules are described in [RFC3261].

10.1. Call-completion status

The cc-state line indicates the CC status of a particular user with an entry in a CC queue. It MUST be present unless the expiration time indicated in the NOTIFY is zero.

```
cc-state = "cc-state" HCOLON ( "queued" / "ready" )
```

The value "queued" indicates that a subscriber's entry in the call-completion queue is not selected for CC recall. The value "ready" indicates that a user's entry in the call-completion queue has been selected for CC recall.

10.2. Call-completion service-retention indication

The service-retention line indicates the support of the retain option. The retain option, if supported at the callee, will maintain the entry in the CC queue, if a CC call has failed due to callee busy condition. If present, this parameter indicates that the retain option is supported, otherwise it is not supported. This parameter

MAY be present in NOTIFY requests.

cc-service-retention = "cc-service-retention" HCOLON "true"

10.3. Call-completion URI

The cc-URI line provides a URI (possibly in the form of a name-addr) which the agent SHOULD use as the request-URI of the CC recall INVITE and the suspend/resume PUBLISH. It SHOULD be provided in all NOTIFYs. The URI SHOULD be globally routable and SHOULD uniquely identify the CCE in question. The syntax provides for generic-params in the value, but this document defines no such parameters. Parameters that are not understood by the subscriber MUST be retained with the URI.

cc-URI = "cc-URI" HCOLON addr-spec

11. Security considerations

The CC facility allows the caller's agent to determine some status information regarding the callee. This information intrinsically diminishes the privacy of the callee; in order to protect sufficiently the privacy of the callee, the overall amount of disclosure must be limited, and the amount of disclosure to any single caller must be limited.

Of course, if a caller is not permitted to call the callee, that caller should not be allowed to establish a CC subscription. Callers that are particularly sensitive about their privacy may reject all CC subscriptions. But in the ordinary case, the optimal protection is to permit any caller to subscribe, but prevent any caller from subscribing for too long, or too often, or in a pattern that does not reveal to the callee (through CC calls) that the subscriptions are taking place.

In legitimate use, CC event subscriptions will be made in stereotyped ways that limit the disclosure of status information:

1. When a subscriber is selected for CC, a call should arrive promptly for the callee, or the subscription should be terminated. This expectation may be violated by a race condition between selection of the subscription for CC and the caller becoming unavailable, but it should be rare that a single subscription will exhibit the race condition more than once.

2. Subscriptions should not remain suspended for longer than the expected duration of a call (a call by the caller to a third

party).

3. Subscriptions should be initiated only shortly after failed incoming calls.

4. Most of the time, a callee should have no queued subscriptions.

Violations of these expectations should be detected by the callee's monitor and reported as possible attempts at privacy violation.

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

In order to prevent Denial of Service (DoS) attacks and manipulations of the call-completion queue by suspending other call-completion entries than the own, a mechanism to correlate the identity of the original caller and the generator of the SUBSCRIBE and PUBLISH request is needed. The RECOMMENDED mechanism to authenticate the identity of the originator of call-completion relevant requests is the SIP Identity mechanism [RFC4474] . Alternatively, CC agents and monitors within an administrative domain or federation of domains MAY use the mechanism described in [RFC3325] to authenticate their identity with a P-Asserted-Identity header field.

Furthermore, the use of presence server functionality to suspend or resume SHOULD be limited to a caller which has an active queue in the callee's monitor. This can be achieved first by monitoring and logging incoming call to the callee and the destination where CC indication was sent, then to ensure subscription to the call completion package is permitted only within a short timeframe after the initial call failed and to only accept PUBLISH request to the presence server functionality if there is an active queue for the caller in question.

Note that regarding the authentication/authorization/billing logic subject to operator policy CC calls or subscriptions do not differ from other basic calls or event subscriptions.

12. IANA considerations

12.1. SIP event package registration for call-completion

This specification registers an event package, based on the registration procedures defined in [RFC6665]. The followings is the information required for such a registration:

Package Name: call-completion

Is this registration for a Template-Package: No.

Published Document: RFC XXXX (Note for RFC Editor: Please fill in XXXX with the RFC number of this specification).

Person and e-mail to contact for further information: Martin Huelsemann, martin.huelsemann@telekom.de

12.2. MIME registration for application/call-completion

MIME media type name: application

MIME subtype name: call-completion

Required parameters: none.

Optional parameters: none.

Encoding considerations: Consists of lines of UTF-8-encoded characters, ended with CR-LF

Security considerations: There are no security considerations internal to the media type. Its typical usage involves the security considerations described in RFC XXXX

(Note for RFC Editor: Please fill in XXXX with the RFC number of this specification).

Interoperability considerations: See RFC XXXX (Note for RFC Editor: Please fill in XXXX with the RFC number of this specification).

Published specification: RFC XXXX (Note for RFC Editor: Please fill in XXXX with the RFC number of this specification)

Applications that use this media type: the implementations of the call-completion features of the Session Initiation Protocol

Additional information:

Magic number(s): none

File extension(s): not expected to be stored in files

Macintosh file type code(s): not expected to be stored in files

Person & email address to contact for further information: Martin Huelsemann, martin.huelsemann@telekom.de

Intended usage: LIMITED USE

Restrictions on usage: none

Author/Change controller: the IETF

12.3. SIP/SIPS URI parameter 'm'

This specification defines one new SIP/SIPS URI parameter 'm' as per the registry created by [RFC3969]. It is used to identify that an INVITE request is a CC call, or to further identify that a SUBSCRIBE request is for the call-completion event package. The parameter may have a value that describes the type of the call-completion operation, as described in this specification.

Name of the Parameter: m

Predefined Values : yes

RFC Reference : [RFC XXXX]

(Note for RFC Editor: Please fill in XXXX with the RFC number of this specification)

12.4. Call-Completion purpose parameter value

This specification adds a new predefined value "call-completion" for the "purpose" header field parameter of the Call-Info header field. This modifies the registry header field parameters and parameter values by adding this RFC as a reference to the line for header field "Call-Info" and parameter name "purpose":

Header Field: Call-Info

Parameter Name: purpose

Predefined Values: yes

Reference: [RFC3261].[RFC5367][[RFC XXXX]]

(Note for RFC Editor: Please fill in XXXX with the RFC number of this

specification)

12.5. 'm' header parameter for Call-Info

This specification extends [RFC3261] to add a new header field parameter 'm' to the Call-Info header field. This adds a row to the registry header field parameters and parameter values:

Header Field: Call-Info

Parameter Name: m

Predefined Values: yes

Reference: [RFC XXXX]

This RFC predefines the values 'BS', 'NR' and 'NL' .

(Note for RFC Editor: Please fill in XXXX with the RFC number of this specification)

13. Acknowledgements

Thanks to Paul Kyzivat, John Elwell, Keith Drage, Andrew Hutton, Thomas Stach, Dennis Luebbers and Christer Holmberg who provided helpful comments, feedback and suggestions.

14. References

14.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3261] 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.

[RFC3515] Sparks, R., "The Session Initiation Protocol (SIP) Refer Method", RFC 3515, April 2003.

[RFC3863] Sugano, H., Fujimoto, S., Klyne, G., Bateman, A., Carr, W., and J. Peterson, "Presence Information Data Format (PIDF)", RFC 3863, August 2004.

[RFC3903] Niemi, A., "Session Initiation Protocol (SIP) Extension for Event State Publication", RFC 3903, October 2004.

[RFC3969] Camarillo, G., "The Internet Assigned Number Authority (IANA) Uniform Resource Identifier (URI) Parameter Registry for the Session Initiation Protocol (SIP)", BCP 99, RFC 3969, December 2004.

[RFC4235] Rosenberg, J., Schulzrinne, H., and R. Mahy, "An INVITE-Initiated Dialog Event Package for the Session Initiation Protocol (SIP)", RFC 4235, November 2005.

[RFC4474] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, August 2006.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC5367] Camarillo, G., Roach, A., and O. Levin, "Subscriptions to Request-Contained Resource Lists in the Session Initiation Protocol (SIP)", RFC 5367, October 2008.

[RFC5627] Rosenberg, J., "Obtaining and Using Globally Routable User Agent URIs (GRUUs) in the Session Initiation Protocol (SIP)", RFC 5627, October 2009.

[RFC6665] Roach, A., "SIP-Specific Event Notification", RFC 6665, July 2012.

14.2. Informative References

[ETS300.356-18]
"Completion of Calls to Busy Subscriber (CCBS) supplementary service", February 1995.

[ITU-T.Q.733]
"DESCRIPTION FOR CALL COMPLETION SUPPLEMENTARY SERVICES USING SS No. 7", February 1995.

[RFC3325] Jennings, C., Peterson, J., and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", RFC 3325, November 2002.

[RFC3725] Rosenberg, J., Peterson, J., Schulzrinne, H., and G. Camarillo, "Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP)",

BCP 85, RFC 3725, April 2004.

[RFC5359] Johnston, A., Sparks, R., Cunningham, C., Donovan, S., and K. Summers, "Session Initiation Protocol Service Examples", BCP 144, RFC 5359, October 2008.

Appendix A. Example Caller's Agent

This section outlines how an autonomous caller's agent can operate using only standard SIP features to interact with the caller's UA. This example is suitable only as a learning aid, as its performance is poor.

The agent monitors calls made from the UA(s) by subscribing to the dialog event package of the UA(s).

The UA(s) or their proxy routes calls made with either of two special dial sequences to the agent, which interprets the INVITEs as indications to make a CC request with BS or NR or NL mode for the latest call made by the UA.

The agent monitors the status of the UA(s) for availability to be used for a CC call by examining the dialog events.

The agent indicates to the UA(s) that CC recall is in progress by making call to the UA(s). If the UA is answered, the agent assumes that the caller is available and plays pre-recorded audio to indicate that CC recall is in progress.

After playing the pre-recorded audio, the caller's agent uses REFER to order the UA to make the CC call to the callee.

Appendix B. Example Callee's Monitor

This section outlines how an autonomous callee's monitor can operate using only standard SIP features to interact with the callee's UA. This example is suitable only as a learning aid, as its performance is poor.

The callee's monitor monitors calls made to the UA(s) by subscribing to the UA(s) dialog events. This enables it to determine their Call-Id's and their final response statuses.

The proxy for the UA(s) routes to the callee's monitor any SUBSCRIBEs for the call-completion event package directed to the URIs serviced by the UA(s).

The callee's monitor monitors the status of the UA(s) to determine when they are in a suitable state to receive a CC call by watching the busy/not-busy status of the UA(s): e.g. a UA is available for CCBS when it is not busy, but a UA is available for CCNR when it becomes not busy after being busy with an established call.

Authors' Addresses

Dale R. Worley
Ariadne Internet Services, Inc.
738 Main St.
Waltham, MA, 02451
US

Phone: +1 781 647 9199
Email: worley@ariadne.com
URI: http://

Martin Huelsemann
Deutsche Telekom
Heinrich-Hertz-Strasse 3-7
Darmstadt, 64307
Germany

Phone: +4961515812765
Email: martin.huelsemann@telekom.de
URI: http://www.telekom.de

Roland Jesske
Deutsche Telekom
Heinrich-Hertz-Strasse 3-7
Darmstadt, 64307
Germany

Phone: +4961515812766
Email: r.jesske@telekom.de
URI: http://www.telekom.de

Denis Alexeitsev
TeleFLASH
Mainzer Landstrasse 47
Frankfurt 60329
Germany

Phone: +49-69-257-378-230
Email: alexeitsev@teleflash.com
URI: http://www.teleflash.com