

**A Call Completion Service for the Session Initiation Protocol (SIP)
Using the Binary Floor Control Protocol (BFCP)
draft-roach-sipping-callcomp-bfcp-00**

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on April 17, 2007.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

This document proposes extensions to the Session Initiation Protocol (SIP) and the Binary Floor Control Protocol (BFCP) for service request and queue manipulation related to the Completion of Calls to Busy Subscribers (CCBS) and Completion of Calls on No Reply (CCNR) services.

Table of Contents

1.	Conventions and Definitions	4
2.	Introduction	4
3.	Requirements	5
3.1.	Service Requirements	5
3.1.1.	Service Capability Indication	5
3.1.2.	Queue Management and Notification	5
3.1.3.	Call Completion Call Priority	6
3.2.	Solution Constraints	6
3.2.1.	PSTN Interoperation	6
3.2.2.	True Native Solution	6
3.2.3.	Protocol Consistency and Explicit Operations	7
4.	Mechanism Overview	7
4.1.	Service Capability Indication	7
4.2.	Queue Management and Notification	7
4.3.	Call Completion Call Priority	8
5.	User Agent Behavior	8
5.1.	Calling Party Behavior	9
5.1.1.	Service Support Detection	9
5.1.2.	Service Activation	9
5.1.3.	Call Re-Attempt	12
5.2.	Called Party Behavior	12
5.2.1.	Service Support Indication	12
5.2.2.	Service Activation	13
5.2.3.	Call Prioritization	15
6.	Protocol Extensions	16
6.1.	SIP "callcomp" Option Tag	16
6.2.	SDP Extensions	16
6.3.	BFCP Extensions	17
6.3.1.	New Message Types	17
6.3.2.	"Paused" Request Status	19
6.3.3.	"TERMINATION-CODE" Extension Attribute	19
7.	Deployment Considerations	20
8.	Examples	21
8.1.	Completion of Call to Busy Subscriber	21
8.2.	Completion of Call on No Response	22
9.	Security Considerations	23
10.	IANA Considerations	24
10.1.	SIP "callcomp" Option Tag	24
10.2.	SDP "service retention" attribute	24
10.3.	BFCP Primitives	24
10.4.	BFCP "Paused" Request Status	24
10.5.	BFCP "TERMINATION-CODE" Extension Attribute	24
10.6.	TERMINATION-CODE Reason Codes Registry	24
11.	References	24
11.1.	Normative References	24
11.2.	Informative References	24

Roach

Expires April 17, 2007

[Page 2]

Author's Address	25
Intellectual Property and Copyright Statements	26

1. Conventions and Definitions

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](#).

2. Introduction

Many legacy phone systems -- including the Public Switched Telephony Network (PSTN) and Private Branch Exchanges (PBXes) -- include services intended to allow calls that have otherwise failed to complete at a later time. The European Telecommunications Standards Institute (ETSI) has defined specific semantics for the behavior of such services, and given them the names "Completion of Calls to Busy Subscribers (CCBS)" and "Completion of Calls on No Reply (CCNR)." For simplicity, this document refers to both services as "Call Completion," except in cases in which a distinction between the two service is useful.

CCBS provides the caller with the ability to complete a requested call to a busy callee without having to make a new call attempt when the callee becomes not busy. The CCBS supplementary service description is defined in ETSI EN 300 357 [TODO: add ref].

CCNR provides the caller with the ability to complete a requested call to a callee without having to make a new call attempt when the callee becomes not busy after having initiated an activity. The CCNR supplementary service description is defined in ETSI EN 301 134 [TODO: add ref].

Typically, the caller initiates a call in the normal fashion for their access device (e.g. dialing a number, selecting an entry from an address book application). When the user receives indication of "ringing" or "busy," they may choose to activate the Call Completion service. The method of activation will vary depending on the user's access device. For a PBX phone, there will typically be a dedicated "callback" button; PC-based clients may have a menu option and/or a hot key; and PSTN terminals will generally activate the service through a set of DTMF tones. The user receives an indication of service activation success or failure.

Once the callee's device becomes available, the caller's phone will indicate that the Call Completion service has been triggered. For PBX phones and PSTN terminals, this will typically be a distinctive ring; for PC-based clients, it may be an audio alert accompanied by a dialog box.

The user responds to this alerting as if he were answering an incoming call. Upon doing so, he receives indication of the far end alerting, and the call proceeds as normal.

At any time before the Call Completion service is triggered by the callee's device becoming available, the caller may suspend or cancel outstanding Call Completion requests.

Of particular note for the Call Completion services as defined by ETSI is the fact that the interaction between multiple callers is rigidly defined. If a single callee has multiple callers waiting for that callee to become available, those callers will receive indication that the callee is available on a service-specific basis (typically, this is first-come-first-served, although other schemes are possible). This aspect of the service requires the maintenance of a queue of waiting callers, as well as a means for callers to manipulate their request in this queue.

If not for this rigidly defined queue behavior, implementation of call completion services could be trivially effected in SIP using existing mechanisms.

3. Requirements

The following sections outline requirements for the Call Completion services. Such requirement are broken into two categories: those that are necessary to implement the SIP Call Completion service in a way that is consistent with the ETSI-defined CCBS and CCNR services; and constraints that interested parties have imposed on the solution to meet their deployment needs.

3.1. Service Requirements

These requirements detail requirements on the SIP Call Completion service necessary to fully emulate the ETSI-defined Call Completion services.

3.1.1. Service Capability Indication

Legacy Call Completion services provide the calling party with an indication that the Call Completion service is available for activation.

3.1.2. Queue Management and Notification

When callers attempt to activate a Call Completion service, they need to be informed whether the activation has succeeded or failed.

Callers also need to be notified of changes in the status of their service request, such as a subsequent failure after successful activation.

Users may need to temporarily suspend their request for a Call Completion service (e.g. due to becoming unavailable or busy). Legacy Call Completion services allow users to perform such request suspensions without losing their position in the Call Completion queue associated with the called party. In order to match such functionality, the SIP Call Completion service must allow queue manipulation capabilities that at least include the ability to suspend and resume queued requests.

3.1.3. Call Completion Call Priority

Legacy Call Completion services prioritize Call Completion calls over normal (non-Call Completion) calls for each called party. In other words, the Call Completion service ensures that the callers who have been waiting for a particular callee are connected to that callee before other potential callers.

3.2. Solution Constraints

These requirements detail constraints on the potential SIP Call Completion service solutions to meet operational requirements of parties interested in the Call Completion service.

3.2.1. PSTN Interoperation

A key motivation in fully replicating the behavior of the ETSI-defined Call Completion services in SIP is the ability to interoperate a SIP Call Completion service with the Legacy Call Completion services defined in existing networks.

In particular, the PSTN-based Call Completion services require the presence of a Call Completion Service Setup (CCSS) indication in any ISUP Initial Address Messages (IAMs) that are sent as a result of the Call-Completion service.

As a result of this requirement, the called SIP user agent must be capable of distinguishing between normal calls and calls that are set up as a result of a Call Completion service.

3.2.2. True Native Solution

Although legacy network interoperation is a key motivator for certain aspects of this solution, a SIP Call Completion service is first and foremost a SIP service. This means that any such solution must be

useful as a stand-alone SIP service.

In general, this means that such a solution should be defined as it applies to native SIP user agents, not as it relates to interoperation with legacy networks.

3.2.3. Protocol Consistency and Explicit Operations

Additionally, as a SIP Call Completion service is a SIP service, it must be consistent with existing SIP protocol mechanisms. Consequently, any requests to activate the service (i.e., insert the caller into the queue) and to pause and unpause the service (i.e., manipulation the callee's queue request) needs to be explicit and consistent with the meaning of the message in which it is carried. In particular, this means that such a solution must not make service activation and manipulation implicit as a side effect of an already well-defined SIP mechanism.

4. Mechanism Overview

The following sections provide a high-level overview of the mechanisms designed to meet the requirements described in [Section 3](#).

4.1. Service Capability Indication

Called parties indicate support for SIP Call Completion services by including the option tag "callcomp" in a "Supported" header field in a response code.

4.2. Queue Management and Notification

The SIP protocol does not presently contain any methods that perform queue manipulation or any similar operations. Consequently, any solution that describes a Call Completion service for SIP in a way that is consistent with existing SIP mechanisms must either define new SIP methods explicitly designed for Call Completion queue manipulation or use the SIP INVITE method to establish a queue-control session.

Historically, new SIP methods have been reserved for functionality that provides a versatile tool that can be applied to a wide variety of problems; for example, the broad REFER method was selected over a proposed TRANSFER method for its wide, non-service-specific application.

The foregoing indicates that there is only one viable approach for the queue management functionality required for a Call Completion

mechanism which meets all the requirements defined in [Section 3](#): the use of INVITE to establish a queue manipulation connection.

Rather than define a new protocol for manipulation of Call Completion queues, this document proposes the re-use of an existing protocol. Fundamentally, Call Completion services can be abstracted to the general problem of multiple users (one or more callers) attempting to access a shared resource (the called party), and being granted access to that resource in a very specific and controlled fashion. This is the precise general problem that the Binary Floor Control Protocol (BFCP), defined in [RFC 4582](#) [TODO: ref].

In particular, when a called party indicates support for the mechanism described in this document, the calling party can activate the Call Completion service by sending a new INVITE request to the called party, containing a "Require: callcomp" header field. This INVITE contains SDP which establishes a BFCP connection. This BFCP connection is then used to add requests to the Call Completion queue (using a FloorRequest message); receive indication of current request status, including indication of when the caller is available (by way of the FloorRequestStatus message); cancel an outstanding request (using the FloorRelease message); and manipulate the caller's request in the queue (using new FloorRequestPause and FloorRequestResume messages).

[4.3.](#) Call Completion Call Priority

In order for the called party to recognize an incoming call as a Call Completion call, called parties provide the calling party with a unique activation URI that the called party can recognize as being associated with the Call Completion service. Although it is perfectly acceptable to generate the URI in other ways, one obvious mechanism that can be used to generate such URIs is through the use of the "grid" parameter on a Globally Routable User-Agent URI (GRUU) [TODO: add ref].

This URI that the called party recognizes as being associated with the call completion service is conveyed to the calling party in the Contact: header field of the response to the BFCP-establishing INVITE request.

[5.](#) User Agent Behavior

Parties to the Call Completion service behave as described in the following subsections.

The calling party MUST send a SIP BYE to terminate the BFCP session

upon receipt of a "FloorRequestStatus" message that indicates a status other than "Pending," "Accepted," or "Granted".

5.1. Calling Party Behavior

The calling part has three primary responsibilities to implement the Call Completion service: detecting support for the Call Completion service, activation of the Call Completion service, and re-attempting to contact the called party once the called party becomes available.

5.1.1. Service Support Detection

Calling parties detect support for the SIP Call Completion service by observing a "Supported: callcomp" header field in any response to an INVITE request. See [Section 6.1](#) for details on the callcomp option tag.

Because INVITE requests may fork, calling parties **MUST** be prepared to receive indication of support for the Call Completion service on several different early dialogs.

For each early dialog established, the calling party's user agent stores the value in the Contact header field, to facilitate activation of the service.

Because the called party needs to receive an indication of service support before activating a Call Completion service, user agents that support the service described in this document **SHOULD** apply the mechanism described in [RFC 3262](#) [TODO: ref] to ensure that provisional responses conveying support for the Call Completion service are delivered reliably.

5.1.2. Service Activation

Service activation consists of two tasks: establishment of a BFCP session; and use of the BFCP session to send queue requests and receive service request status.

5.1.2.1. BFCP Session Establishment

Once the user indicates the desire to activate the Call Completion service, the calling party sends INVITE requests to each and every URI collected from Contact header fields during the procedure described in [Section 5.1.1](#). These INVITE requests contain "Required: callcomp" header fields.

The calling party uses the procedures described in [RFC 4583](#) [TODO: ref] to establish one or more BFCP connections with the called party.

Calling parties MUST NOT send media sections in such INVITE requests to establish media other than BFCP.

To avoid any doubt about the setting of attributes in the SDP:

- o The calling party indicates "c-only" in the "floorctrl" attribute associated with the BFCP media line.

Additionally, this specification adds a new attribute to be associated with BFCP m-lines:

- o A new "service-retention" attribute. The valid values for this attribute are "true" and "false." If this attribute is not present, it is presumed to be "false." The called party includes the "service-retention" attribute to indicate support of the "retain" option defined by ETSI. The retain option, if supported by both the calling party and the called party, will maintain the Call Completion request in the destination B queue, if a Call Completion call has failed due to destination busy condition. [TODO: this should be explained more clearly]

Establishment of the BFCP sessions does not activate the Call Completion service. Activation of the service itself is performed via BFCP messages.

5.1.2.2. BFCP Usage

Once one or more BFCP sessions are established, the calling party will send BFCP messages to activate, deactivate, pause, and resume the Call Completion service. The syntax and semantics of these requests are described in [RFC 4582](#).

In the context of the Call Completion service, the "floor" corresponds to permission to place a Call Completion call to the called party.

Consequently, the calling party activates the Call Completion service by sending a "FloorRequest" message on each of the established BFCP sessions.

Similarly, the calling party learns the status of the Call Completion service request by receiving "FloorRequestStatus" messages. The state of the request can be determined by examining the status field of these floor status requests:

Pending: The activation request has been received, but has not necessarily succeeded yet.

Accepted: The activation request has succeeded. If non-zero, the "Queue Position" field will indicate the user's position in the call completion queue.

Paused: The activation request has been suspended at the calling party's request. If non-zero, the "Queue Position" field will indicate the user's position in the call completion queue.

Granted: The called party is now available. The calling party user agent attempts to establish a call between the calling party and the called party as described in [Section 5.1.3](#). If the call is not placed quickly enough, the calling party may receive another "FloorRequestStatus" message with a status of "Accepted," which indicates that the user has been placed back in the Call Completion queue.

Denied: The called party has refused the request to activate the Call Completion service. If present, the STATUS-INFO attribute contains additional human-readable information, such as why the request was denied. If present, the TERMINATION-CODE attribute (defined in [Section 6.3.3](#)) may give additional machine-interpretable information about the nature of the denial.

Canceled: The Call Completion request has been canceled, either at the calling party's request, or due to some administrative reason, such as a service timeout. If present, the TERMINATION-CODE attribute (defined in [Section 6.3.3](#)) may give additional machine-interpretable information about the nature of the cancellation.

To pause or resume the Call Completion service request without giving up their location in the Call Completion queue, the calling party uses the "FloorRequestPause" and "FloorRequestResume" messages, respectively.

To cancel the Call Completion service, the calling party sends a "FloorRelease" message. The calling party also sends a "FloorRelease" message once the called party has been successfully contacted.

If the called party terminates the BFCP session unexpectedly, the calling party MUST consider the Call Completion service activation request to be canceled.

5.1.3. Call Re-Attempt

When the calling party receives a "FloorRequestStatus" message with a status of "Granted," it MUST alert its user of the called party's availability. When the calling party indicates that they are ready for the call to proceed, the calling party user agent initiates the call. It does so by sending an INVITE request to the URI that it received in the Contact header field of the 200-class response to the INVITE that set up the BFCP session.

Once the call succeeds, the calling party user agent SHOULD cancel any other outstanding Call Completion requests that relate to the same call attempt.

5.2. Called Party Behavior

The called party has three primary responsibilities to implement the Call Completion service: indicating support for the Call Completion service, activation of the Call Completion service, and prioritization of Call Completion calls over ordinary calls.

5.2.1. Service Support Indication

If a called party supports the SIP Call Completion service described in this document, it MUST include "Supported: callcomp" all non-100 INVITE provisional responses (i.e. 101 through 199) and in any other response that indicates that the called party's endpoint is temporarily unavailable. Called parties MAY include such an indication in other responses.

Because the called party needs to receive such an indication before activating a Call Completion service, user agents that support the service described in this document SHOULD send a non-100 provisional response immediately upon receipt of an INVITE request, even if a final response will be sent immediately. If no other provisional response is appropriate, the 183 (Session Progress) response code can be used.

Any provisional responses containing a "Supported: callcomp" header field MUST also contain a Contact header field that is guaranteed to route back to the same user agent.

Additionally, because the called party needs to receive an indication of service support before activating a Call Completion service, user agents that support the service described in this document SHOULD apply the mechanism described in [RFC 3262](#) [TODO: ref] to ensure that provisional responses conveying support for the Call Completion service are delivered reliably.

5.2.2. Service Activation

Service activation consists of two tasks: establishment of a BFCP session; and use of the BFCP session to receive queue requests and update the calling party about service request status.

5.2.2.1. BFCP Session Establishment

If an INVITE containing "Require: callcomp" is received by a user agent supporting this specification (assuming the request is acceptable to the user agent), it immediately responds by sending a 200-class response to accept the session.

The 200-class response to such a request MUST also contain a Contact header field that is both guaranteed to route back to the same user agent and which the user agent recognizes as being associated with the Call Completion service. This MAY be the same URI that was returned in the original INVITE provisional response, but it is not required to be the same.

The called party uses the procedures described in [RFC 4583](#) [TODO: ref] to establish a BFCP connection from the calling party to the called party. If the SDP in an INVITE with "Require: callcomp" contains m= lines other than those used to establish BFCP connections, the called party MUST reject the INVITE with a 488 (Not Acceptable here) response.

To avoid any doubt about the setting of attributes in the SDP:

- o The called party indicates "s-only" in the "floorctrl" attribute associated with the BFCP media line.
- o The called party sets the "confid" and "userid" attributes to any value that it desires. These values will be reflected back to the called party in subsequent BFCP messages.
- o The called party sets the "floorid" attribute to any value that it desires. These values will be reflected back to the called party in subsequent BFCP messages. The "floorid" attribute will not contain an "mstrm" portion, since the floor is not associated with any media streams.

Additionally, this specification adds a new attribute to be associated with BFCP m-lines:

- o A new "service-retention" attribute. The valid values for this attribute are "true" and "false." If this attribute is not present, it is presumed to be "false." The called party includes the "service-retention" attribute to indicate support of the "retain" option defined by ETSI. The retain option, if supported by both the calling party and the called party, will maintain the Call Completion request in the destination B queue, if a Call Completion call has failed due to destination busy condition. [TODO: this should be explained more clearly]

Establishment of the BFCP session does not activate the Call Completion service. Activation of the service itself is performed via BFCP messages.

5.2.2.2. BFCP Usage

Once the BFCP session is established, the called party will receive BFCP messages from the calling party intended to activate, deactivate, pause, and resume the Call Completion service. The syntax and semantics of these requests are described in [RFC 4582](#).

In the context of the Call Completion service, the "floor" corresponds to permission to place a Call Completion call to the called party.

Consequently, the called party activates the Call Completion service upon receipt of a "FloorRequest" message from the calling party.

Similarly, the called party conveys the status of the Call Completion service request by sending the calling party "FloorRequestStatus" messages.

Upon receipt of a "FloorRequest," the called party will respond with a "FloorRequestStatus" containing a status of "Pending."

Once the Call Completion service activation has been successfully completed, the called party sends an additional "FloorRequestStatus" with a status of "Accepted." If the called party wishes to share such information with the calling party, this "FloorRequestStatus" message MAY also contain a "Queue Position" field that indicates the calling party's position in the Call Completion queue. The called party MAY send new "FloorRequestStatus" messages with "Accepted" status from time to time to update the calling party regarding their position in the queue.

When the calling party reaches the front of the Call Completion queue (i.e., is permitted to re-attempt their original call), the called

party sends a "FloorRequestStatus" message with a status of "Granted." If the calling party does not place the call re-attempt quickly enough, the called party may place that caller back in the queue and send another "FloorRequestStatus" message containing a status of "Accepted." Determination of what constitutes "fast enough" and where in the queue the calling party is placed are implementation-dependent.

If the called party denies the calling party's request to activate the Call Completion service, then the called party sends a "FloorRequestStatus" message with a status of "Denied." The STATUS-INFO attribute MAY be used to convey additional human-readable information, such as why the request was denied.

If the calling party's request fails due to an administrative reason -- such as a service timeout -- then the called party sends a "FloorRequestStatus" with a status of "Canceled."

If the called party sends a "FloorRequestStatus" with a status of either "Denied" or "Canceled", it MAY include a TERMINATION-CODE attribute, as defined in [Section 6.3.3](#).

If the called party receives a "FloorRequestPause" message, it should suspend the calling party's progression through the queue, and send the calling party a "FloorRequestStatus" message with a status of "Paused." While in the "Paused" state, a calling party's position in the queue will not switch to "Granted" until the calling party sends a "FloorRequestResume" message. Upon receipt of a "FloorRequestResume" message, the called party will resume progression of the calling party through the Call Completion queue.

Upon receipt of a "FloorRelease" message, the called party can consider the Call Completion service activation to be canceled or completed (depending on whether the calling party successfully contacted the called party). The called party MUST send a SIP BYE to terminate the BFCP session upon receipt of a "FloorRelease" message.

If the calling party terminates the BFCP session unexpectedly, the called party can also consider the Call Completion service activation request to be canceled.

[5.2.3](#). Call Prioritization

When the called party user agent receives an INVITE request sent to a URI that is associated with a Call Completion attempt, it first checks whether the calling party matches the user that is currently considered "Active" in its call completion queue. If the user does not match, the user agent replies with a 480 (Temporarily

Unavailable) response.

If the user does match, then the called party user agent alerts its user, and allows the call to proceed as normal.

If a called party user agent receives an INVITE request sent to a URI that is not related to the Call Completion service while it has users in its Call Completion queue, it responds with a 480 (Temporarily Unavailable) response.

6. Protocol Extensions

Extensions to existing protocols necessary to support the mechanism described in this document are minimal. The following sections detail minor extensions to SIP, SDP, and BFCP.

6.1. SIP "callcomp" Option Tag

This specification uses the SIP option tag mechanism for negotiating support for the Call Completion service. Refer to [RFC 3261](#) [TODO: ref] for the normative description of processing of the "Supported" and "Require" header fields.

In practice, this specification's use of the the option tag mechanism guarantees success except in the most degenerate corner cases: UACs never indicate that support for the "callcomp" option tag is required unless they have already contacted the UAS to which the request is addressed, and already received indication that the UAS supports the "callcomp" extension.

Use of "Require: callcomp" in INVITE messages meant to establish BFCP sessions used to control the Call Completion service is included to satisfy the [RFC 3261](#) requirement that a UAC MUST insert a Require header field into a request if the UAC wishes to insist that a UAS understand an extension in order to process the request. Because the INVITE would not be usable without applying the extension defined in this document, the calling party is obligated to include it in such INVITE requests.

6.2. SDP Extensions

This document makes use of the SDP extensions defined in [RFC 4583](#) [TODO: add ref]. In addition to these extensions, This specification adds one more attribute which is applicable to BFCP-establishing m= lines when they are used to set up Call Completion BFCP sessions.

The Augmented BNF notation which defined this new attribute is as

follows:

```
sr-attribute = "a=service-retention:" sr-value
sr-value     = "true" / "false"
```

6.3. BFCP Extensions

This document adds mechanisms to BFCP that allow the pausing and unpausing of floor control requests that have not yet been granted.

The sections that follow define these extensions in a generic fashion that apply to all usages of BFCP. In practice, their use in the SIP Call Completion service will be limited to use on a single floor at a time.

6.3.1. New Message Types

This specification defines two new message types ("Primitives") for the Binary Floor Control Protocol defined by [RFC 4582](#) [TODO: ref]. The new primitive types are as follows; these extend Table 1 in [RFC 4582](#) [TODO: ref]:

Value	Primitive	Direction
20	FloorRequestPause	P -> S
21	FloorRequestResume	P -> S

S: Floor Control Server
P: Floor Participant

6.3.1.1. FloorRequestPause message

Floor participants use the FloorRequestPause message to suspend pending floor requests. While paused, floor requests remain in the floor control queue; however, they will not enter the "Granted" state until the floor participant resumes the floor request using a "FloorRequestResume" message. The following is the format of the FloorRequestPause message:

```
FloorRequestPause = (COMMON-HEADER)
                   (FLOOR-REQUEST-ID)
                   *[EXTENSION-ATTRIBUTE]
```

The floor participant uses the FLOOR-REQUEST-ID that was received in the response to the FloorRequest message that the FloorRequestPause

message is suspending.

Note that if the floor participant requested several floors as an atomic operation (i.e., in a single FloorRequest message), all the floors are suspended as an atomic operation as well (i.e., all are suspended at the same time).

A message from the floor control server is considered a response to the FloorRequestPause message if the message from the floor control server has the same Conference ID, Transaction ID, and User ID as the FloorRequestPause message, as described in [RFC 4582](#) [TODO: ref].

If the response is a FloorRequestStatus message, the Request Status value in the OVERALL-REQUEST-STATUS attribute (within the FLOOR-REQUEST-INFORMATION grouped attribute) will be Paused. If the response is an Error message, the floor control server could not process the FloorRequestPause message for some reason, which is described in the Error message.

[6.3.1.2](#). FloorRequestResume message

Floor participants use the FloorRequestResume message to resume a floor control request that had previously been suspended by use of a FloorRequestPause message. The following is the format of the FloorRequestResume message:

```
FloorRequestResume = (COMMON-HEADER)
                    (FLOOR-REQUEST-ID)
                    *[EXTENSION-ATTRIBUTE]
```

The floor participant uses the FLOOR-REQUEST-ID that was received in the response to the FloorRequest message that the FloorRequestResume message is resuming.

Note that if the floor participant requested several floors as an atomic operation (i.e., in a single FloorRequest message), all the floors are resumed as an atomic operation as well (i.e., all are resumed at the same time).

A message from the floor control server is considered a response to the FloorRequestResume message if the message from the floor control server has the same Conference ID, Transaction ID, and User ID as the FloorRequestResume message, as described in [RFC 4582](#) [TODO: ref].

If the response is a FloorRequestStatus message, the Request Status value in the OVERALL-REQUEST-STATUS attribute (within the FLOOR-REQUEST-INFORMATION grouped attribute) will be Active or Granted. If the response is an Error message, the floor control server could not

The "Family" field defines a group of reason codes specific to a particular usage of the TERMINATION-CODE attribute. This specification defines reasons in family 1. Other specifications may make use of other families.

servers can be co-located with the proxy, so that this proprietary mechanism never crosses a network. Such co-location also allows the proxy to prevent "normal" calls from getting through when callers are queued in the Call Completion queue].

8. Examples

[TODO: the examples need to be expanded with message details]

8.1. Completion of Call to Busy Subscriber

Caller	Callee
(1) INVITE caller	
----->	
	180 contains Contact with GRUU
	pointing to this terminal. We
	will call this GRUU "[a]"
(2) 180	
<-----	
(3) PRACK	
----->	
(4) 200 (PRACK)	
<-----	
(5) 486 Busy	
<-----	
(6) ACK	
----->	
	INVITE sets up BFCP session
(7) INVITE [a]	
----->	
(8) 200 (INVITE) (contact=[a])	
<-----	
(9) ACK	
----->	
(10) BFCP connection	
.....	
(11) FloorRequest	
----->	
(12) FloorRequestStatus (Pending)	
<-----	
	Callee accepts CCBS request
(13) FloorRequestStatus (Accepted)	
<-----	
	Called party becomes available
(14) FloorRequestStatus (Granted)	
<-----	

Roach

Expires April 17, 2007

[Page 21]

		Calling party is alerted and
		initiates call completion
(15) INVITE [a]		
----->		
(16) 180		
<-----		
(17) PRACK		
----->		
(18) 200 (PRACK)		
<-----		
(19) 200 (INVITE)		
<-----		
(20) ACK		
----->		
(21) FloorRelease		
----->		
(22) BYE (BFCP session)		
<-----		
(23) 200		
----->		
(24) Media		
.....		

8.2. Completion of Call on No Response

Caller	Callee
(1) INVITE caller	
----->	
	180 contains Contact with GRUU
	pointing to this terminal. We
	will call this GRUU "[a]"
(2) 180	
<-----	
(3) PRACK	
----->	
(4) 200 (PRACK)	
<-----	
(5) CANCEL	
----->	
(6) 200 (CANCEL)	
<-----	
(7) 487 Request Terminated	
<-----	
(8) ACK	
----->	
	INVITE sets up a BFCP session

Roach

Expires April 17, 2007

[Page 22]

```

| (9) INVITE [a] |
| -----> |
| (10) 200 (INVITE) (contact=[a]) |
| <----- |
| (11) ACK |
| -----> |
| (12) BFCP connection |
| ..... |
| (13) FloorRequest |
| -----> |
| (14) FloorRequestStatus (Pending) |
| <----- |
| | Callee accepts CCBS request
| (15) FloorRequestStatus (Accepted) |
| <----- |
| | Callee becomes available
| (16) FloorRequestStatus (Granted) |
| <----- |
| | Calling party is alerted and
| | initiates call completion
| (17) INVITE [a] |
| -----> |
| (18) 180 |
| <----- |
| (19) PRACK |
| -----> |
| (20) 200 (PRACK) |
| <----- |
| (21) 200 (INVITE) |
| <----- |
| (22) ACK |
| -----> |
| (23) FloorRelease |
| -----> |
| (24) BYE (BFCP session) |
| <----- |
| (25) 200 |
| -----> |
| (26) Media |
| ..... |

```

9. Security Considerations

[TODO: not that it's not important, just that I'm out of time]

10. IANA Considerations

10.1. SIP "callcomp" Option Tag

[TODO: fill this in]

10.2. SDP "service retention" attribute

[TODO: fill this in]

10.3. BFCP Primitives

[TODO: fill this in]

10.4. BFCP "Paused" Request Status

[TODO: fill this in]

10.5. BFCP "TERMINATION-CODE" Extension Attribute

[TODO: fill this in]

10.6. TERMINATION-CODE Reason Codes Registry

[TODO: fill this in]

11. References

11.1. Normative References

11.2. Informative References

Author's Address

Adam Roach
Estacado Systems
17210 Campbell Rd.
Suite 250
Dallas, TX 75252
US

Phone: sip:adam@estacado.net
Email: adam@estacado.net

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM 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.

Copyright Statement

Copyright (C) The Internet Society (2006). This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

Acknowledgment

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

