DISPATCH Working Group
Internet Draft
Intended status: Standards Track
Expires: February 20, 2011

H. Kaplan
Acme Packet

August 20, 2010

A Session Initiation Protocol (SIP) INFO Package for Dual-Tone Multi-Frequency (DTMF) Events
draft-kaplan-dispatch-info-dtmf-package-00

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and 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 February 20, 2011.

Copyright and License Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the BSD License.

Abstract

The SIP INFO request method now supports explicit indication and exchange of specified application information. Such usages are documented as an "INFO Package", following the requirements outlined in [info-packages]. This document specifies one such SIP INFO Package, for the purpose of indicating DTMF signals.

Table of Contents

1. Introduction....2
2. Terminology....3
3. The "dtmf" INFO Package....3
3.1. Overall Description....3
4. Overview of Operation....3
5. INFO Package Definition....4
5.1. INFO Package Name....4
5.2. INFO Bodies....4
5.3. UA Behavior....4
6. Example Exchange....5
7. Security Considerations....6
8. IANA Considerations....6
9. References....6
9.1. Normative References....6
9.2. Informative References....7
Author's Address....7

1. Introduction

Dual-Tone Multi-Frequency tones are used in numerous telephony applications, in multiple ways and for multiple purposes. From a SIP protocol perspective, they follow one or both of two paths: the RTP media path, and/or the SIP signaling path. DTMF in the media path is handled by [RFC4733], with a reasonable level of interoperability, if both ends of the session support it.

There are, however, numerous cases in which DTMF tones need to be sent in the signaling path. The most often cited use-case for such is calling-card applications, where the calling-card application server is not in the media path but needs to detect DTMF tones. There are many other use-cases, however, including SIP-to-H.323 Interworking Gateways, distributed IVR and voicemail-retrieval systems, and some IP-PBX systems.

Historically there have been multiple ways of exchanging DTMF in the SIP signaling path: INFO, NOTIFY and KPML [RFC4730]. KPML works by having the application server(s) create Subscriptions to the end User-Agent (UA), and the UA sends NOTIFY messages within the

SUBSCRIBE dialog to indicate the DTMF tones. To date, KPML has seen limited deployment. Another method, using non-KPML NOTIFY requests, has usually been implemented by sending NOTIFY messages in the INVITE-based dialog, without a SUBSCRIBE; but the use of NOTIFY as such is not very common.

Legacy INFO usage for DTMF is widely deployed, but has no documented standard and there are at least two known variants in use (although one is arguably far more popular). The two variants use different MIME bodies in the INFO message to indicate the DTMF: an "application/dtmf" and an "application/dtmf-relay" MIME body type, with dtmf-relay being the most commonly used one. Although no standard exists for dtmf-relay, it is documented on several websites and extremely simple, with a high level of interoperability. This document is intended to standardize such a mechanism, using the INFO Package model, for a "dtmf" INFO Package.

2. 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 RFC 2119. The terminology in this document conforms to RFC 2828, "Internet Security Glossary".

3. The "dtmf" INFO Package

This document defines a new INFO Package called "dtmf", for the purpose of exchanging DTMF tone signals in the SIP signaling path.

3.1. Overall Description

The dtmf INFO Package is used to carry DTMF tone signals, indicating the specific tone and duration, in "application/dtmf-relay" type MIME bodies in INFO requests messages inside an INVITE-based dialog.

4. Overview of Operation

The general concept is that the UAC and UAS negotiate support for a "dtmf" INFO Package during the initial INVITE transaction, using the [info-package] mechanism. Including the "dtmf" INFO Package name in the Recv-Info header field means the UA sending the header supports receiving DTMF events using this mechanism. When the far-end has indicated that the it supports receiving "dtmf", and the user presses a DTMF digit, the UA sends it in an INFO request. The digit pressed and the duration it was pressed for are encoded in an "application/dtmf-relay" MIME type attachment in the INFO, described later in this document.

If a SIP server in the signaling path between the calling UAC and answering UAS wants to receive DTMF indications following this mechanism, they must act as a B2BUA. Such behavior is out of scope of this document.

5. INFO Package Definition

5.1. INFO Package Name

This document defines a SIP INFO Package as defined in [info-packages]. The INFO Package token name for this package is "dtmf".

5.2. INFO Bodies

Applications using this INFO Package MUST include an "application/dtmf-relay" body in INFO requests to indicate which digit was pressed by the user. The body contains exactly two lines: one of the button pushed, the other of the duration. The body is described in ABNF form as follows:

```
Dtmf-relay-body = digit-line CRLF duration-line

digit-line      = "Signal" EQUAL SP button
button          = DIGIT / "A" / "B" / "C" / "D" / "*" / "#"

duration-line   = "Duration" EQUAL SP msecs
msecs           = 1*4(DIGIT)  ;100-5000 millisecs
```

5.3. UA Behavior

A UA supporting this draft MUST indicate the user-pressed button through INFO if the remote UA indicated it supports receiving the "dtmf" INFO Package, per the rules in [info-packages]. If [RFC4733] (i.e., RTP-based DTMF events) was also indicated by the far-end in SDP, and the local UA supports sending such, it MUST send the event indication through both means simultaneously. If the UA also supports [KPML] and some entity subscribed for the "kpml" package for the same call, the UA still MUST send dtmf indication through the INFO, and MUST also send such through a [KPML] Notify assuming it would have done so otherwise. (i.e., assuming the regex matched and so on)

The UA MUST populate the "application/dtmf-relay" body, as defined earlier, with the button pressed and the duration it was pressed for. Technically, this actually requires the INFO to be generated when the user *releases* the button, however if the user has still not released a button after 5 seconds, which is the maximum duration

supported by this mechanism, the UA should generate the INFO at that time.

6. Example Exchange

In the following example, Alice initiates a call to Bob. Alice can support sending or receiving "dtmf" events.

Alice generates the following: (note: much has been left out for simplicity)

```
INVITE sip:bob@example.com SIP/2.0
Via: SIP/2.0/UDP 192.0.2.1:5060;branch=z9hG4bKnashds10
To: Bob <sip:bob@example.com>
From: Alice <sip:alice@example.net>;tag=1234567
Call-Id: 123456mcmxcix
CSeq: 1 INVITE
Contact: <sip:alice@192.0.2.1>
Accept: application/sdp, application/dtmf-relay
Recv-Info: dtmf
```

Bob checks the headers, and can support receiving "dtmf".

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 192.0.2.1:5060;branch=z9hG4bKnashds10
To: Bob <sip:bob@example.com>;tag=abcdefg
From: Alice <sip:alice@example.net>;tag=1234567
Call-Id: 123456mcmxcix
CSeq: 1 INVITE
Accept: application/sdp, application/dtmf-relay
Recv-Info: dtmf
```

Since he sent the Recv-Info header in the 180, Bob also sends it in the 200.

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.0.2.1:5060;branch=z9hG4bKnashds10
To: Bob <sip:bob@example.com>;tag=abcdefg
From: Alice <sip:alice@example.net>;tag=1234567
Call-Id: 123456mcmxcix
CSeq: 1 INVITE
Contact: <sip:bob@192.0.2.2>
Accept: application/sdp, application/dtmf-relay
Recv-Info: dtmf
```

At some later time, Alice presses number 6 on her keypad, for 100ms. She sends the following:

```
INFO sip:bob@192.0.2.2 SIP/2.0
Via: SIP/2.0/UDP 192.0.2.1:5060;branch=z9hG4bKnabcdef
From: Alice <sip:alice@example.net>;tag=1234567
To: Bob <sip:bob@example.com>;tag=abcdefg
Call-Id: 123456mcmxcix
CSeq: 2 INFO
Contact: <sip:alice@192.0.2.1>
Info-Package: dtmf
Content-Disposition: Info-Package
Content-Type: application/dtmf-relay
Content-Length: 26

Signal= 6
Duration= 100
```

7. Security Considerations

There are no specific security issues for this mechanism, beyond those already applicable to SIP-based session signaling and [info-packages].

8. IANA Considerations

This document will presumably register the INFO Package name "dtmf" and "application/dtmf-relay" MIME type, if it moves forward.

9. References

9.1. Normative References

[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.

[RFC2976] Donovan, S., "The SIP INFO Method", RFC 2976, October 2000.

[info-packages] Holmberg, C., Burger, E., Kaplan, H., "Session Initiation Protocol (SIP) INFO Method and Package Framework", draft-ietf-sipcore-info-events-08, May 2010.

9.2. Informative References

[KPML] Burger, E., Dolly, M., "A Session Initiation Protocol (SIP) Event Package for Key Press Stimulus (KPML)", RFC 4730, November 2006

[4733] Schulzrinne, H., Taylor, T., "RTP Payload for DTMF Digits, Telephony Tones, and Telephony Signals", RFC 4733, December 2006

Author's Address

Hadriel Kaplan
Acme Packet
100 Crosby Drive
Bedford, MA 01703

Email: hkaplan@acmepacket.com

DISPATCH WG
Internet-Draft
Intended status: Informational
Expires: January 13, 2011

A. Romanow
Cisco
S. Botzko
Polycom
July 12, 2010

Problem Statement for Telepresence Multi-streams
draft-romanow-dispatch-telepresence-prob-statement-01.txt

Abstract

Telepresence systems create a "being there" conferencing experience. A number of issues need to be solved largely by manipulating multiple audio and video streams. Different systems take different approaches, employ different techniques, and convey information by using different vocabularies, making interoperability extremely challenging. This problem statement describes the typical issues that must be solved and uses examples to illustrate the kind of diversity that makes interworking problematic.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on January 13, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
2. Terminology 4
3. Fundamental Issues for Telepresence 4
4. Manipulating Media Streams 5
5. Examples of Interworking Issues 6
5.1. Designating Roles and Positions for transmitted streams . 6
5.2. Multipoint 7
5.3. Capability Negotiation 9
5.4. Differences in Media Characteristics 9
5.4.1. Aspect Ratio 9
5.4.2. Visual Scale 11
6. IANA Considerations 12
7. Security Considerations 12
8. Acknowledgements 12
9. Informative References 13
Authors' Addresses 13

1. Introduction

In a Telepresence conference, the idea is to create a feeling of presence - that you are in the same room with the remote parties. In order to create the "being there" or telepresence experience, a number of technical issues need to be solved. These issues are addressed by manipulating multiple media streams, video and audio - by describing them, controlling them, and signaling about them. The fundamental features of telepresence require handling multiple streams of media, and considering additional characteristics of those streams beyond those normally specified in existing videoconferencing standards.

Different telepresence systems approach solving the basic issues differently. They use disparate techniques, and they describe, control and signal media in dissimilar fashions. Such diversity creates an interoperability problem. The same issues are solved in different ways by different systems, so that they are not directly interoperable. This makes interworking difficult at best and sometimes impossible.

Some degree of interworking is possible through transcoding and translation. This requires additional devices, which are expensive and not entirely automatic. Specialized knowledge is required to operate a telepresence conference where the endpoints use different equipment and a transcoding and translating device is employed for interoperability. Often such conferences are interrupted by difficulties that arise.

The general problem that needs to be solved is this. The transmitting side sends audio and video streams based upon a model for rendering a realistic depiction from this information. If the receiving side belongs to the same vendor, it works with the same model and renders the information according to that shared model. However, if the receiver and the sender are from different vendors, the models they each have for rendering presence differ.

It is as if Alice and Bob are at different sites. Alice needs to tell Bob information about what her camera and sound equipment see at her site so that Bob’s receiver can create a display that will capture the important characteristics of her site. Alice and Bob need to agree on what the salient characteristics are as well as how to represent and communicate them. The telepresence multi-steam work seeks to describe the sender situation in a way that allows the receiver to render it realistically though it may have a different rendering model than the sender.

This problem statement identifies the fundamental issues that need to

be addressed to provide telepresence in typical use case scenarios. We show how different approaches to solving the problems and different techniques for handling multiple media create a challenge for interoperability.

This document describes some of the problems that arise, it is not an complete list, but rather it is more illustrative than exhaustive. Requirements, use cases and solutions are discussed in other documents.

2. 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 RFC 2119 [RFC2119].

3. Fundamental Issues for Telepresence

The fundamental issues that must be handled to produce a typical telepresence conference, either point to point or multipoint include:

1. Participant display
    A. Placement of video
    B. Size
    C. Angle
    D. Overlap
    E. Display technology
2. Audio
    A. Placement, emanating from right place
    B. Type of audio
3. Different number of screens on sender and receiver sides
4. Participant display for multipoint
    A. Placement of video

B. Continuous presence

C. Control of display, how does it change? - automatic, user

5. Maintaining eye contact and gaze connection

6. Panoramic view for site switching

7. Mismatches between media characteristics between sender and receiver, such as:

A. aspect ratio

B. format

C. frame rate

D. resolution

8. Presentation

A. What methodology?

9. Security

A. SRTP?

B. Key methodology

4. Manipulating Media Streams

In addressing the fundamental issues, multiple media streams are handled in the following ways:

1. Sender and receiver understand each others capabilities

A. Number of video, audio and presentation streams that can be sent/received simultaneously

B. What media signaling protocol being used (SDP, proprietary, etc.)

2. Streaming control

3. Feedback mechanisms

4. Signaling about RTP payload

5. Media control signaling

   A. Video refresh

   B. Flow control

6. Signaling media formats and media capabilities

7. Signaling content type

8. Signaling device type

9. Signaling network characteristics per stream

10. Floor control signaling

5. Examples of Interworking Issues

This section describes several examples that illustrate the kinds of incompatibilities that arise when different systems take different approaches to an issue.

5.1. Designating Roles and Positions for transmitted streams

Senders and receivers need to have the same vocabulary and understanding of stream roles and positions in order to place them appropriately. For example one system may define roles as: center, left, right, legacy center, legacy right, legacy left, auxiliary 1/5 fps and auxiliary 30 fps positions. These roles as defined are a combination of "input devices" + "codec type/format" for transmission positions, and a combination of "stream decoders/output devices" + "codec type/format" for receive positions. Another system will not have the exact same vocabulary and meaning, though it still has to accomplish the same placement task.

How the cameras and encoders are wired determines how the local scene is displayed on the remote screen. In many systems right and left need to be exchanged to be seen properly, but this depends on the way the equipment is wired.

In describing how to display the local scene, the language can be misleading if there is no agreed upon reference for right and left. [for example, more]

Although often the video is displayed on separate monitors, it is

also possible to use projectors to create a video wall. In this case, there may be an overlap region between cameras which allows for projector blending. Also, although cameras are generally arranged to create a seamless panoramic view of the participants, it is also possible for there to be gaps between cameras (and corresponding gaps between displays).

There is also no reference for image size. Some rooms use proportionally larger displays, and set the camera field of view to show participants either standing or sitting at life size. Others use smaller displays, and set the field of view for sitting participants (cropping off heads when people stand). In order to preserve full size display when these systems interoperate, both systems must rescale their video.

5.2. Multipoint

Multipoint conferences, where there are more than two endpoints, create a wealth of technical issues to be solved. The primary one is which participants to display on each screen at each site. If the number of sites is greater than can be shown on the number of displays at a site, this adds to the complexity. There are, of course, almost unlimited ways this can be handled. We discuss the common approaches and how they differ.

The local screens can show all the camera image from the a particular remote site (site switching); or each local screen can show a participant or two from each of the remote sites (segment switching); or local displays can show a composite of remote camera shots (continuous presence). The choice of who to display on a screen can be determined by users, or, more often, automated according to voice activity level.

[Add user-controlled personal telepresence scenario.]

Policies are created and implemented in many ways. They tend to be based on some combination of what H.323 defines as centralized and decentralized. One of the challenges is that the endpoints in the conference may have different number of cameras and displays from each other so a common mode on the number of streams and their priority is required. Also, the various endpoints might have different bandwidth constraints and support different codec profiles.

A centralized multipoint conference is one in which all participating endpoints communicate in a point-to-point fashion with an MCU. The endpoints transmit their control, audio, video, and/or data streams to the MCU. The MCUA centrally manages the conference, processes the audio, video and/or data streams, and returns the processed streams

to each endpoint. In this mode, the MCU will mix the audio streams; and if using centralized video, will either use voice activated video switch, where everyone will see the active speaker and the speaker will see the previous speaker, or will use continuous presence mode, where the MCU will create a video stream with sub windows for each of the participants. MCUs can support multiple video layouts and they can be created automatically based on the number of participants or by a conference management application.

There are three methods commonly used for video stream distribution in centralized multipoint conferences. The three conference policies above can be implemented using any of these technologies.

Simple video switching (forwarding) has the advantage of low latency and low complexity. It can be used if all systems are capable of receiving the encodings used by the sending endpoints (including both the video codec and the image resolution/aspect ratio). In some situations it can be wasteful of bandwidth.

Full video transcoding usually has higher latency than switching It does not require system to be capable of receiving identical encodings, and different sites can connect with different bandwidths.

Layered video encoding combines some of the benefits of video switching and video transcoding. It is more complex than video switching, but less complex than video transcoding. Bandwidth and resolution can be reduced for each site. Since this is done by filtering out layers of the original encoding, the available bandwidths and resolutions are not as fine-grained as full video transcoding.

In decentralized mode or full mesh mode each endpoint creates its display mode. This requires each endpoint to receive multiple streams and send its video and audio to all participants, using multicast of unicast.

In practice, multicast is not now being used in commercial systems, so the size of a strictly decentralized multipoint conference is limited.

There are analogous issues for audio. Like video, the audio is rotated, so there is no clarity on the meaning of left and right. Since the number of streams, microphones, and speakers are not matched, the systems need to re-process the received audio in order to create the correct sound field for their respective rooms.

There are two ways in which the audio might be handled in this use case:

- o A single stereo audio stream is sent to the remote site, just as in standard videoconferencing.

- o Three monaural audio streams are sent to the remote site, with proprietary signaling to associate each audio stream with a video stream.

Microphones and speakers positions vary; and there is no agreed upon way to describe their placement. There is no agreed upon reference for audio level. In addition, audio may be sent as an independent stream from each microphone or as a multi-channel channel stream.

5.3. Capability Negotiation

Call setup for the telepresence conference will start with a single call establishing one video media stream. After the connection is established, a proprietary capability negotiation takes place that will enable both sides to identify that they are telepresence applications and capable of having two more video sessions and provide the connectivity information. The result is that two or more video sessions are established. The system may use two new SIP call legs or just add the two new video streams to the existing dialog.

[more to be added]

5.4. Differences in Media Characteristics

Media characteristics such as video format, aspect ratio, and visual scale can be handled differently at different sites creating incompatibility. To interwork, an adaptive strategy is necessary. Although differences in media characteristic must also be handled in a typical video conference, the problem is made more complex in Telepresence due to the multiple screens, cameras and streams.

Two examples - aspect ratio and visual scale are described here.

5.4.1. Aspect Ratio

If the aspect ratios in different sites are not the same, some technique needs to be applied to adjust for the difference. Although the same situation arises in normal video conferencing, multiple streams in telepresence conferencing causes more difficulties.

For simplicity let us assume a point to point case - two conference room on a point to point call. Both rooms have 3 screens and 3 cameras, as in 4.1 above. Both rooms have identical visual scale - the display width and distance between the participants and the displays are identical in both rooms. However the equipment -

cameras and displays - in each room has a different aspect ratio, 16:9 in one room and 4:3 in the other.

Although 4:3 is usually associated with standard definition TV and 16:9 with HDTV, telepresence systems may choose the aspect ratio to obtain a particular field of view. Projecting images in the 16:9 aspect ratio offers a wider presentation angle that shows fine details well (the pixel density is greater than a 4:3 system of the same resolution and scale). In the room with 16:9 media characteristic, people are shown at full size when they are seated. However, when they stand up the height of the display results in their image being cropped so that their heads are not shown. The other room uses projectors to display HD images with 4:3 aspect ratios. This results in an increased image height - the vertical field of view is 33% greater than the 16:9 system. The increased height allows most of the population to be shown full size whether they are standing or sitting.

Some strategy is necessary to deal with the case of the two sites having a point to point call. In order to convert formats of unequal ratios a variety of techniques can be used, such as: zooming (enlarging) and cropping (removing), letterboxing (adding horizontal bars), pillarboxing (adding vertical bars) to retain the original format’s aspect ratio, or scaling (which distorts) in a variety of ways.

For the video sent from the 4:3 room to the 16:9 room, several techniques can be used:

1. The 16:9 system might simply crop the top 1/4 of each 4:3 image. This will result in full size display, eye contact, and gaze awareness for the individuals who are seated. However, the standing presenter’s head will be cropped.

2. The 16:9 system might stretch each to the 4:3 images to fully fit the 16:9 display. This would reduce image height (creating geometric distortion) and create eye-contact error. Continuity of the panoramic image would be preserved.

3. The 16:9 system could pillarbox each of the 4:3 images, placing horizontal borders on the three displays. This results in reducing the image size to less than full size. It also destroys the continuity of the panoramic image, and introduces additional error in eye contact and gaze awareness.

4. The 16:9 system could pillarbox only the center display. This reduces the size of the presenter who is the focus of the meeting.

5. The 16:9 system could also crop the bottom of the center display. Visually this reduces the height of the presenter, but maintains full size. There is a vertical discontinuity in the panoramic image. Whether this is objectionable or not depends on the room layout.

Strategies 4 and 5 could be accomplished in response to a user command or automatically. The details will be discussed in more detail in future documents.

For the video sent from the 16:9 room to the 4:3 room, the receiving system simply letterboxes the video displays. Since the scales are identical, this full size image displays in the 4:3 room.

For the video sent from the 16:9 room to the 4:3 room, the common techniques are:

1. The 4:3 system places the border above the image. This maintains eye contact for those who are seated, but cannot maintain eye contact for the presenter.

2. The 4:3 system places the border below the images. If the 16:9 system crops the bottom of the center display then this will maintain eye contact for the presenter and the remote site.

3. The 4:3 system centers the images. Eye contact suffers for everyone, but the worst case eye contact error is better controlled.

In this use case, negotiation between the systems is not strictly necessary, no matter which scheme is used. However, the best user experience is obtained if both systems have knowledge about apect ratios being used and which participants are standing and which are sitting so they can adjust optimally.

5.4.2. Visual Scale

The visual scale of displays may differ between sites. Again, let us use the point to point case as a simple example. Assume two conference rooms in a point to point call. One room is designed for 6 participants, and has three 16:9 screens and 3 cameras. This room is designed to show participants at their normal size when seated (2 participants per camera/display). It does not have adequate display height to capture those who are standing. The second room is also designed for 6 participants, but shows 3 participants per camera/ display also at their full size. Therefore, it only needs two 16:9 cameras/display pairs. Since the field of view in both the vertical and horizontal is increased by 50%, it also shows those who are

standing without cropping.

For the video sent from the 2 screen (larger scale) room to the 3 screen (smaller scale) room, two approaches can be used:

1. The 3 screen system might simply show the participants on two of its displays. Participants will be shown at 67% of their full size. Eye contact and gaze awareness will be lost.

2. The 3 screen system might construct and display a vertically cropped 3-screen view, showing 2 participants on each screen. Participants will be shown at full size, with preservation of eye contact and gaze awareness.

For the video sent from the 3 screen to the 2 screen room, there are two analogous approaches:

1. The 2 screen system selects 2 streams and simply shows them on its displays. Participants will be shown at 150% of their normal size. Eye contact and gaze awareness will be lost, and some of the remote site is lost.

2. The 2 screen system might construct and display a 2 screen view (with a vertical border on the top) which shows 3 participants on each screen. Participants will be shown at full size, with preservation of eye contact and gaze awareness.

Although there is no need for negotiation between the systems, the best user experience is obtained if both systems have knowledge of the visual scale, and where individuals are seated, and can then choose the best manner of display.

6. IANA Considerations

This document contains no IANA considerations.

7. Security Considerations

While there are likely to be security considerations for any solution for telepresence interoperability, this document has no security considerations.

8. Acknowledgements

The draft has benefitted from input from a number of people including

Roni Even, Jim Cole, Nermeen Ismail, Nathan Buckles.

9. Informative References

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

Authors' Addresses

Allyn Romanow
Cisco
San Jose, CA 95134
US

Email: allyn@cisco.com

Stephen Botzko
Polycom
Andover, MA 01810
US

Email: stephen.botzko@polycom.com

DISPATCH WG
Internet-Draft
Intended status: Informational
Expires: January 13, 2011

A. Romanow
Cisco
S. Botzko
M. Duckworth
Polycom
R. Even
Gesher Erove
T. Eubanks
Iformata Communications
July 12, 2010

Use Cases for Telepresence Multi-streams
draft-romanow-dispatch-telepresence-use-cases-01.txt

Abstract

Telepresence conferencing systems seek to create the sense of really being present. A number of techniques for handling audio and video streams are used to create this experience. When these techniques are not similar, interoperability between different systems is difficult at best, and often not possible. Conveying information about the relationships between multiple streams of media would allow senders and receivers to make choices to allow telepresence systems to interwork. This memo describes the most typical and important use cases for sending multiple streams in a telepresence conference.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on January 13, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
2. Terminology 3
3. Telepresence Scenarios Overview 4
4. Use Case Scenarios 6
4.1. Point to point meeting: symmetric 7
4.2. Point to point meeting: asymmetric 7
4.3. Multipoint meeting 9
4.4. Presentation 10
4.5. Multipoint Education Usage 11
4.6. Other 12
5. Acknowledgements 13
6. IANA Considerations 13
7. Security Considerations 13
8. Informative References 13
Authors' Addresses 13

1. Introduction

Telepresence applications try to provide a "being there" experience for conversational video conferencing. Often this telepresence application is described as "immersive telepresence" in order to distinguish it from traditional video conferencing, and from other forms of remote presence not related to conversational video conferencing, such as avatars and robots. The salient characteristics of telepresence are often described as: full-sized, immersive video, preserving interpersonal interaction and allowing non-verbal communication.

Although telepresence systems are based on open standards such as RTP [RFC3550], SIP [RFC3261] , H.264, and the H.323 suite of protocols, they cannot easily interoperate with each other without operator assistance and expensive additional equipment which translates from one vendor to another. A standard way of describing the multiple streams constituting the media flows and the fundamental aspects of their behavior, would allow telepresence systems to interwork.

This draft presents a set of use cases describing typical scenarios. Requirements will be derived from these use cases in a separate document. The use cases are described from the viewpoint of the users. They are illustrative of the user experience that needs to be supported. It is possible to implement these use cases in a variety of different ways. A problem statement draft describes the difficulties when one participant’s equipment has a different approach than another’s.

Many different scenarios need to be supported. Our strategy in this document is to describe in detail the most common and basic use cases. These will cover most of the requirements. Additional scenarios that bring new features and requirements will be added.

We look at telepresence conferences that are point-to-point and multipoint. In some settings, the number of displays is similar at all sites, in others, the number of displays differs at different sites. Both cases are considered. Also included is a use case describing display of presentation or content.

The document structure is as follows: Section 2 presents the document terminology, Section 3 gives an overview of the scenarios, and Section 4 describes use cases.

2. 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 RFC 2119 [RFC2119].

3. Telepresence Scenarios Overview

This section describes the general characteristics of the use cases and what the scenarios are intended to show. The typical setting is a business conference, which was the initial focus of telepresence. Recently consumer products are also being developed. We specifically do not include in our scenarios the infrastructure aspects of telepresence, such as room construction, layout and decoration.

Telepresence systems are typically composed of one or more video cameras and encoders and one or more display monitors of large size (around 60"). Microphones pick up sound and audio codec(s)produce one or more audio streams. The cameras used to present the telepresence users we will call participant cameras (and likewise for displays). There may also be other cameras, such as for document display. These will be referred to as presentation or content cameras, which generally have different formats, aspect ratios, and frame rates from the participant cameras. The presentation videos may be shown on participant screen, or on auxiliary display screens. A user’s computer may also serve as a virtual content camera, generating an animation or playing back a video for display to the remote participants.

We describe such a telepresence system as sending M video streams, N audio streams, and D content streams to the remote system(s). (Note that the number of audio streams is generally not the same as the number of video streams.)

The fundamental parameters describing today’s typical telepresence scenario include:

1. The number of participating sites
2. The number of visible seats at a site
3. The number of cameras
4. The number of audio channels
5. The screen size
6. The display capabilities - such as resolution, frame rate, aspect ratio

7. The arrangement of the displays in relation to each other

8. Similar or dissimilar number of primary screens at all sites

9. Type and number of presentation displays

10. Multipoint conference display strategies - for example, the camera-to-display mappings may be static or dynamic

11. The camera viewpoint

12. The cameras fields of view and how they do or do not overlap

The basic features that give telepresence its distinctive characteristics are implemented in disparate ways in different systems. Currently Telepresence systems from diverse vendors interoperate to some extent, but this is not supported in a standards based fashion. Interworking requires that translation and transcoding devices be included in the architecture. Such devices increase latency, reducing the quality of interpersonal interaction. Use of these devices is often not automatic; it frequently requires substantial manual configuration and a detailed understanding of the nature of underlying audio and video streams. This state of affairs is not acceptable for the continued growth of telepresence - we believe telepresence systems should have the same ease of interoperability as do telephones.

There is no agreed upon way to adequately describe the semantics of how streams of various media types relate to each other. Without a standard for stream semantics to describe the particular roles and activities of each stream in the conference, interoperability is cumbersome at best.

In a multiple screen conference, the video and audio streams sent from remote participants must be understood by receivers so that they can be presented in a coherent and life-like manner. This includes the ability to present remote participants at their true size for their apparent distance, while maintaining correct eye contact, gesticular cues, and simultaneously providing a spatial audio sound stage that is consistent with the video presentation.

The receiving device that decides how to display incoming information needs to understand a number of variables such as the spatial position of the speaker, the field of view of the cameras; the camera zoom; which media stream is related to each of the displays; etc. It is not simply that individual streams must be adequately described, to a large extent this already exists, but rather that the semantics of the relationships between the streams must be communicated. Note

that all of this is still required even if the basic aspects of the streams, such as the bit rate, frame rate, and aspect ratio, are known. Thus, this problem has aspects considerably beyond those encountered in interoperation of single-node video conferencing units.

4. Use Case Scenarios

Our development of use cases is staged, initially focusing on what is currently typical and important. Use cases that add future or more specialized features will be added later as needed. Also, there are a number of possible variants for these use cases, for example, the audio supported may differ at the end points (such as mono or stereo versus surround sound), etc. These issues will be discussed in more depth in the problem statement document.

The use cases here are intended to be hierarchical, in that the earlier use cases describe basics of telepresence that will also be used by later use cases.

Many of these systems offer a full conference room solution where local participants sit on one side of a table and remote participants are displayed as if they are sitting on the other side of the table. The cameras and screens are typically arranged to provide a panoramic (left to right) view of the remote room.

The sense of immersion and non-verbal communication is fostered by a number of technical features, such as:

1. Good eye contact, which is achieved by careful placement of participants, cameras and screens.

2. Camera field of view and screen sizes are matched so that the images of the remote room appear to be full size.

3. The left side of each room is presented on the right display at the far end; similarly the right side of the room is presented on the left display. The effect of this is that participants of each site appear to be sitting across the table from each other. If two participants on the same site glance at each other, all participants can observe it. Likewise, if a participant on one site gestures to a participant on the other site, all participants observe the gesture itself and the participants it includes.

4.1. Point to point meeting: symmetric

In this case each of the two sites has an identical number of screens, with cameras having fixed fields of view, and one camera for each screen. The sound type is the same at each end. As an example, there could be 3 cameras and 3 screens in each room, with stereo sound being sent and received at each end.

The important thing here is that each of the 2 sites has the same number of screens. Each screen is paired with a corresponding camera. Each camera / screen pair is typically connected to a separate codec, producing a video encoded stream for transmission to the remote site, and receiving a similarly encoded stream from the remote site.

Each system has one or multiple microphones for capturing audio. In some cases, stereophonic microphones are employed. In other systems, a microphone may be placed in front of each participant (or pair of participants). In typical systems all the microphones are connected to a single codec that sends and receives the audio streams as either stereo or surround sound. The number of microphones and the number of audio channels are often not the same as the number of cameras. Also the number of microphones is often not the same as the number of loudspeakers.

The audio may be transmitted as multi-channel (stereo/surround sound) or as distinct and separate monophonic streams. Audio levels should be matched, so the sound levels at both sites are identical. Loudspeaker and microphone placements are chosen so that the sound "stage" (orientation of apparent audio sources) is coordinated with the video. That is, if a participant on one site speaks, the participants at the remote site perceive her voice as originating from her visual image. In order to accomplish this, the audio needs to be mapped at the received site in the same fashion as the video. That is, audio received from the right side of the room needs to be output from loudspeaker(s) on the left side at the remote site, and vice versa.

4.2. Point to point meeting: asymmetric

In this case, each site has a different number of screens and cameras than the other site. The important characteristic of this scenario is that the number of displays is different between the two sites. This creates challenges which are handled differently by different telepresence systems.

This use case builds on the basic scenario of 3 screens to 3 screens. Here, we use the common case of 3 screens and 3 cameras at one site,

and 1 screen and 1 camera at the other site, connected by a point to point call. The display sizes and camera fields of view at both sites are basically similar, such that each camera view is designed to show two people sitting side by side. Thus the 1 screen room has up to 2 people seated at the table, while the 3 screen room may have up to 6 people at the table.

The basic considerations of defining left and right and indicating relative placement of the multiple audio and video streams are the same as in the 3-3 use case. However, handling the mismatch between the two sites of the number of displays and cameras requires more complicated maneuvers.

For the video sent from the 1 camera room to the 3 screen room, usually what is done is to simply use 1 of the 3 displays and keep the second and third displays inactive, or put up the date, for example. This would maintain the "full size" image of the remote side.

For the other direction, the 3 camera room sending video to the 1 screen room, there are more complicated variations to consider. Here are several possible ways in which the video streams can be handled.

1. The 1 screen system might simply show only 1 of the 3 camera images, since the receiving side has only 1 screen. Two people are seen at full size, but 4 people are not seen at all. The choice of which 1 of the 3 streams to display could be fixed, or could be selected by the users. It could also be made automatically based on who is speaking in the 3 screen room, such that the people in the 1 screen room always see the person who is speaking. If the automatic selection is done at the sender, the transmission of streams that are not displayed could be suppressed, which would avoid wasting bandwidth.

2. The 1 screen system might be capable of receiving and decoding all 3 streams from all 3 cameras. The 1 screen system could then compose the 3 streams into 1 local image for display on the single screen. All six people would be seen, but smaller than full size. This could be done in conjunction with reducing the image resolution of the streams, such that encode/decode resources and bandwidth are not wasted on streams that will be downsized for display anyway.

3. The 3 screen system might be capable of including all 6 people in a single stream to send to the 1 screen system. For example, it could use PTZ (Pan Tilt Zoom) cameras to physically adjust the cameras such that 1 camera captures the whole room of six people. Or it could recompose the 3 camera images into 1 encoded stream

to send to the remote site. These variations also show all six people, but at a reduced size.

4. Or, there could be a combination of these approaches, such as simultaneously showing the speaker in full size with a composite of all the 6 participants in smaller size.

The receiving telepresence system needs to have information about the content of the streams it receives to make any of these decisions. If the systems are capable of supporting more than one strategy, there needs to be some negotiation between the two sites to figure out which of the possible variations they will use in a specific point to point call.

4.3. Multipoint meeting

In a multipoint telepresence conference, there are more than two sites participating. Additional complexity is required to enable media streams from each participant to show up on the displays of the other participants.

Clearly, there are a great number of topologies that can be used to display the streams from multiple sites participating in a conference.

One major objective for telepresence is to be able to preserve the "Being there" user experience. However, in multi-site conferences it is often (in fact usually) not possible to simultaneously provide full size video, eye contact, common perception of gestures and gaze by all participants. Several policies can be used for stream distribution and display: all provide good results but they all make different compromises.

One common policy is called site switching. Let’s say the speaker is at site A and everyone else is at a "remote" site. When the room at site A shown, all the camera images from site A are forwarded to the remote sites. Therefore at each receiving remote site, all the screens display camera images from site A. This can be used to preserve full size image display, and also provide full visual context of the displayed far end, site A. In site switching, there is a fixed relation between the cameras in each room and the displays in remote rooms. The room or participants being shown is switched from time to time based on who is speaking or by manual control, e.g., from site A to site B.

Segment switching is another policy choice. Still using site A as where the speaker is, and "remote" to refer to all the other sites, in segment switching, rather than sending all the images from site A,

only the speaker at site A is shown. The camera images of the current speaker and previous speakers (if any) are forwarded to the other sites in the conference. Therefore the screens in each site are usually displaying images from different remote sites - the current speaker at site A and the previous ones. This strategy can be used to preserve full size image display, and also capture the non-verbal communication between the speakers. In segment switching, the display depends on the activity in the remote rooms - generally, but not necessarily based on audio / speech detection).

A third possibility is to reduce the image size so that multiple camera views can be composited onto one or more screens. This does not preserve full size image display, but provides the most visual context (since more sites or segments can be seen). Typically in this case the display mapping is static, i.e., each part of each room is shown in the same location on the display screens throughout the conference.

Other policies and combinations are also possible. For example, there can be a static display of all screens from all remote rooms, with part or all of one screen being used to show the current speaker at full size.

4.4. Presentation

In addition to the video and audio streams showing the participants, additional streams are used for presentations.

In systems available today, generally only one additional video stream is available for presentations. Often this presentation stream is half-duplex in nature, with presenters taking turns. The presentation video may be captured from a PC screen, or it may come from a multimedia source such as a document camera, camcorder or a DVD. In a multipoint meeting, the presentation streams for the currently active presentation are always distributed to all sites in the meeting, so that the presentations are viewed by all.

Some systems display the presentation video on a screen that is mounted either above or below the three participant screens. Other systems provide monitors on the conference table for observing presentations. If multiple presentation monitors are used, they generally display identical content. There is considerable variation in the placement, number, and size or presentation displays.

In some systems presentation audio is pre-mixed with the room audio. In others, a separate presentation audio stream is provided (if the presentation includes audio).

In H.323 systems, H.239 is typically used to control the video presentation stream. In SIP systems, similar control mechanisms can be provided with BFCP [RFC4582]. These mechanisms are suitable for managing a single presentation stream.

Although today's systems remain limited to a single video presentation stream, there are obvious uses for multiple presentation streams.

1. Frequently the meeting convener is following a meeting agenda, and it is useful for her to be able to show that agenda to all participants during the meeting. Other participants at various remote sites are able to make presentations during the meeting, with the presenters taking turns. The presentations and the agenda are both shown, either on separate displays, or perhaps re-scaled and shown on a single display.

2. A single multimedia presentation can itself include multiple video streams that should be shown together. For instance, a presenter may be discussing the fairness of media coverage. In addition to slides which support the presenter's conclusions, she also has video excerpts from various news programs which she shows to illustrate her findings. She uses a DVD player for the video excerpts so that she can pause and reposition the video as needed. Another example is an educator who is presenting a multi-screen slide show. This show requires that the placement of the images on the multiple displays at each site be consistent.

There are many other examples where multiple presentation streams are useful.

4.5. Multipoint Education Usage

The importance of this example is that the multiple video streams are not used to create an immersive conferencing experience with panoramic views at all the site. Instead the multiple streams are dynamically used to enable full participation of remote students in a university class. In some instances the same video stream is displayed on multiple displays in the room, in other instances an available stream is not displayed at all.

The main site is a university auditorium which is equipped with three cameras. One camera is focused on the professor at the podium. A second camera is mounted on the wall behind the professor and captures the class in its entirety. The third camera is co-located with the second, and is designed to capture a close up view of a questioner in the audience. It automatically zooms in on that

student using sound localization.

Although the auditorium is equipped with three cameras, it is only equipped with two screens. One is a large screen located at the front so that the class can see it. The other is located at the rear so the professor can see it. When someone asks a question, the front screen shows the questioner. Otherwise it shows the professor (ensuring everyone can easily see her).

The remote sites are typical immersive telepresence room with three camera/screen pairs.

All remote sites display the professor on the center screen at full size. A second screen shows the entire classroom view when the professor is speaking. However, when a student asks a question, the second screen shows the close up view of the student at full size. Sometimes the student is in the auditorium; sometimes the speaking student is at another remote site. The remote systems never display the students that are actually in that room.

If someone at the remote site asks a question, then the screen in the auditorium will show the remote student at full size (as if they were present in the auditorium itself). The display in the rear also shows this questioner, allowing the professor to see and respond to the student without needing to turn her back on the main class.

When no one is asking a question, the screen in the rear briefly shows a full-room view of each remote site in turn, allowing the professor to monitor the entire class (remote and local students). The professor can also use a control on the podium to see a particular site - she can choose either a full-room view or a single camera view.

Realization of this use case does not require any negotiation between the participating sites. Endpoint devices (and an MCU if present) - need to know who is speaking and what video stream includes the view of that speaker. The remote systems need some knowledge of which stream should be placed in the center. The ability of the professor to see specific sites (or for the system to show all the sites in turn) would also require the auditorium system to know what sites are available, and to be able to request a particular view of any site. Bandwidth is optimized if video that is not being shown at a particular site is not distributed to that site.

4.6. Other

Additional use cases will be added in the future.

Add a typical case with mixture of immersive telepresence and legacy systems, including telephony only.

5. Acknowledgements

The draft has benefitted from input from a number of people including Alex Eleftheriadis, Tommy Andre Nyquist, Mark Gorzynski, Charles Eckel, Nermeen Ismail, Mary Barnes, Jim Cole.

6. IANA Considerations

This document contains no IANA considerations.

7. Security Considerations

While there are likely to be security considerations for any solution for telepresence interoperability, this document has no security considerations.

8. Informative 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.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC4582] Camarillo, G., Ott, J., and K. Drage, "The Binary Floor Control Protocol (BFCP)", RFC 4582, November 2006.

Authors' Addresses

Allyn Romanow
Cisco
San Jose, CA 95134
US

Email: allyn@cisco.com

Stephen Botzko
Polycom
Andover, MA 01810
US

Email: stephen.botzko@polycom.com

Mark Duckworth
Polycom
Andover, MA 01810
US

Email: mark.duckworth@polycom.com

Roni Even
Gesher Erove
Tel Aviv,
Israel

Email: ron.even.tlv@gmail.com

Marshall Eubanks
Iformata Communications
Dayton, Ohio 45402
US

Email: marshall.eubanks@ilformata.com

Dispatch Working Group
Internet-Draft
Intended status: Informational
Expires: May 3, 2012

C. Eckel
T. Kristensen
M. Thompson
G. Sandbakken
E. McLeod
Cisco

October 31, 2011

Revision of the Binary Floor Control Protocol (BFCP) for use over an unreliable transport
draft-sandbakken-dispatch-bfcp-udp-03

Abstract

This draft describes how to extend the Binary Floor Control Protocol (BFCP) for use over an unreliable transport. It details the differences from the BFCP protocol definition document and the Session Description Protocol (SDP) format specified for BFCP streams.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on May 3, 2012.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 4
2. Terminology 4
3. Motivation 4
3.1. Alternatives Considered 6
3.1.1. ICE TCP 6
3.1.2. Teredo 6
3.1.3. GUT 7
3.1.4. UPnP IGD 7
3.1.5. NAT PMP 7
4. Difference from RFC4582 8
4.1. Overview of Operation (4) 8
4.1.1. Floor Participant to Floor Control Server Interface (4.1) 8
4.2. COMMON-HEADER Format (5.1) 8
4.3. ERROR-CODE (5.2.6) 10
4.4. FloorRequestStatusAck (5.3.14) 10
4.5. ErrorAck (5.3.15) 11
4.6. FloorStatusAck (5.3.16) 11
4.7. Goodbye (5.3.17) 11
4.8. GoodbyeAck (5.3.18) 12
4.9. Transport (6) 12
4.9.1. Reliable Transport (6.1) 12
4.9.2. Unreliable Transport (6.2) 13
4.9.2.1. Congestion Control 15
4.9.2.2. ICMP Error Handling 15
4.9.3. Large Message Considerations 15
4.10. Lower-Layer Security (7) 16
4.11. Protocol Transactions (8) 17
4.12. Server Behavior (8.2) 17
4.13. Timers (8.3) 17
4.14. Request Retransmission Timer, T1 (8.3.1) 17
4.15. Response Retransmission Timer, T2 (8.3.2) 18
4.16. Timer Values (8.3.3) 18
4.17. Authentication and Authorization (9) 18
4.17.1. TLS Based Mutual Authentication (9.1) 19
4.18. Receiving a Response [to a FloorRequest Message] (10.1.2) 19
4.19. Receiving a Response [to a FloorRelease Message] (10.2.2) 19
4.20. Receiving a Response [to a ChairAction Message] (11.2) . . 19
4.21. Receiving a Response [to a FloorQuery Message] (12.1.2) . 19

4.22. Receiving a Response [to a FloorRequestQuery Message] (12.2.2) 19
4.23. Receiving a Response [to a UserQuery Message] (12.3.2) . . 20
4.24. Receiving a Response [to a Hello Message] (12.4.2) 20
4.25. Reception of a FloorRequestStatus Message (13.1.3) 20
4.26. Reception of a FloorStatus Message (13.5.3) 20
4.27. Reception of an Error Message (13.8.1) 20
4.28. Security Considerations (14) 21
4.29. IANA Considerations - Primitive Subregistry (15.2) 21
4.30. IANA Considerations - Error Code Subregistry (15.4) . . . 21
4.31. Example Call Flows for BFCP over Unreliable Transport (Appendix A) 21
5. Revision of RFC4583 25
5.1. Fields in the 'm' Line (3) 25
5.2. Authentication (8) 25
5.3. Security Considerations (10) 26
5.4. Registration of SDP 'proto' Values (11.1) 26
6. NAT Traversal 26
7. Future Work 26
8. Acknowledgements 27
9. References 27
9.1. Normative References 27
9.2. Informative References 28
Appendix A. Change History 29
A.1. draft-sandbakken-dispatch-bfcp-udp-02 to -03 29
A.2. draft-sandbakken-dispatch-bfcp-udp-01 to -02 29
A.3. draft-sandbakken-dispatch-bfcp-udp-00 to -01 30
A.4. draft-sandbakken-xcon-bfcp-udp-02 to draft-sandbakken-dispatch-bfcp-udp-00 30
A.5. draft-sandbakken-xcon-bfcp-udp-01 to -02 31
A.6. draft-sandbakken-xcon-bfcp-udp-00 to -01 32
Authors' Addresses 32

1. Introduction

This draft describes how to extend the BFCP protocol to support unreliable transport. Minor changes to the transaction model are introduced in that all requests now have an appropriate response to complete the transaction. The requests are sent with a retransmit timer associated with the response to achieve reliability.

This extension does not change the semantics of BFCP. It permits UDP as an alternate transport. Existing implementations, in the spirit of the approach detailed in earlier versions of this draft (see Appendix A), have demonstrated the approach to be feasible. Initial compatibility among implementations has been achieved at previous interoperability events. The purpose of this draft is to formalize and publish the extension from the standard specification to facilitate complete interoperability between implementations.

The content of this draft relates to the BFCP protocol specification [RFC4582] and the SDP format for describing BFCP streams [RFC4583]. This draft is written with the goal of identifying the extensions associated with adding support for UDP as an alternate transport to an existing BFCP implementation.

2. 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].

3. Motivation

In existing video conferencing deployments, BFCP is used to manage the floor for the content sharing associated with the conference. For peer to peer scenarios, including business to business conferences and point to point conferences in general, it is frequently the case that one or both endpoints exists behind a NAT/ firewall. BFCP roles are negotiated in the offer/answer exchange as specified in [RFC4583], resulting in one endpoint being responsible for opening the TCP connection used for the BFCP communication.

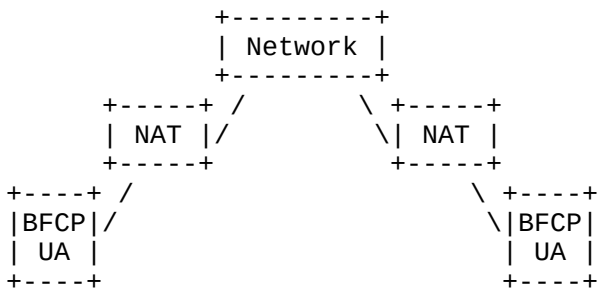


Figure 1: Use Case

The communication session between the video conferencing endpoints typically consists of a number of RTP over UDP media streams, for audio and video, and a BFCP connection for floor control. Existing deployments are most common in, but not limited to, enterprise networks. In existing deployments, NAT/firewall traversal for the RTP streams works using ICE and/or other methods, including those described in [I-D.ietf-mmusic-media-path-middleboxes].

When enhancing an existing SIP based video conferencing deployment with support for content sharing, the BFCP connection often poses a problem. The reasons for this fall into two general classes. First, there may be a strong preference for UDP based signaling in general. On high capacity endpoints (e.g. PSTN gateways or SIP/H.323 interworking gateways), TCP can suffer from head of line blocking, and it uses many kernel buffers. Network operators view UDP as a way to avoid both of these. Second, establishment and traversal of the TCP connection involving ephemeral ports, as is typically the case with BFCP over TCP, can be problematic, as described in Appendix A of [I-D.ietf-mmusic-ice-tcp]. A broad study of NAT behavior and peer-to-peer TCP establishment for a comprehensive set of TCP NAT traversal techniques over a wide range of commercial NAT products concluded it was not possible to establish a TCP connection in 11% of the cases [IMC05]. The results are worse when focusing on enterprise NATs. A study of hole punching as a NAT traversal technique across a wide variety of deployed NATs reported consistently higher success rates when using UDP than when using TCP [P2PNAT].

To overcome the problems with establishing TCP flows between BFCP entities, this draft defines UDP as an alternate transport for BFCP, leveraging the same mechanisms in place for the RTP over UDP media streams for the BFCP communication. When using UDP as the transport, it is RECOMMENDED to follow the guidelines provided in [RFC5405]. NAT traversal for BFCP over UDP entities is discussed in more detail in Section 6.

The authors view this extension as an admittedly non-ideal, but pragmatic, solution to an existing deployment challenge.

3.1. Alternatives Considered

In selecting the approach of defining UDP as an alternate transport for BFCP, several alternatives were considered and explored to some degree. Each of these is discussed briefly in the following subsections. In summary, while these alternatives work in a number of scenarios, they are not sufficient, in and of themselves, to address the use case targeted by this draft.

3.1.1. ICE TCP

ICE TCP [I-D.ietf-mmusic-ice-tcp] extends ICE to TCP based media, including the ability to offer a mix of TCP and UDP based candidates for a single stream. ICE TCP has, in general, a lower success probability for enabling TCP connectivity without a relay if both of the hosts are behind a NAT (see Appendix A of [I-D.ietf-mmusic-ice-tcp]) than enabling UDP connectivity in the same scenarios. The happens because many of the currently deployed NATs in video conferencing networks do not support the flow of TCP hand shake packets seen in case of TCP simultaneous-open, either because they do not allow incoming TCP SYN packets from an address to which a SYN packet has been sent to recently, or because they do not properly process the subsequent SYNACK. Implementing various techniques advocated for candidate collection in [I-D.ietf-mmusic-ice-tcp] should increase the success probability, but many of these techniques require support from some network elements (e.g., from the NATs). Such support is not common in enterprise firewalls and NATs.

3.1.2. Teredo

Teredo [RFC4380] enables nodes located behind one or more IPv4 NATs to obtain IPv6 connectivity by tunneling packets over UDP. Teredo extensions [RFC6081] provide additional capabilities to Teredo, including support for more types of NATs and support for more efficient communication.

As defined, Teredo could be used to make BFCP work for the video conferencing use cases addressed in this draft. However, running the service requires the help of "Teredo servers" and "Teredo relays" [RFC4380]. These servers and relays generally do not exist in the existing video conferencing deployments. It also requires IPv6 awareness on the endpoints. It should also be noted that ICMP6, as used with Teredo to complete an initial protocol exchange and confirm that the appropriate NAT bindings have been set up, is not a conventional feature of IPv4 or even IPv6, and some currently

deployed IPv6 firewalls discard ICMP messages. As these networks continue to evolve and tackle the transaction to IPv6, Teredo servers and relays may be deployed, making Teredo available as a suitable alternative to BFCP over UDP.

3.1.3. GUT

GUT [I-D.manner-tsvwg-gut] attempts to facilitate tunneling over UDP by encapsulating the native transport protocol and its payload (in general the whole IP payload) within a UDP packet destined to the well-known port GUT_P. Unfortunately, it requires user-space TCP, for which there is not a readily available implementation, and creating one is a large project in itself. This draft has expired and its future is still not clear as it has not yet been adopted by a working group.

3.1.4. UPnP IGD

Universal Plug and Play Internet Gateway Devices (UPnP IGD) sit on the edge of the network, providing connectivity to the Internet for computers internal to the LAN, but do not allow Internet devices to connect to computers on the internal LAN. IGDs enable a computer on an internal LAN to create port mappings on their NAT, through which hosts on the Internet can send data that will be forwarded to the computer on the internal LAN. IGDs may be self-contained hardware devices or may be software components provided within an operating system.

In considering UPnP IGD, several issues exist. Not all NATs support UPnP, and many that do support it are configured with it turned off by default. NATs are often multilayered, and UPnP does not work well with such NATs. For example, a typical DSL modems acts as a NAT, and the user plugs in a wireless access point behind that, which adds another layer NAT. The client can discover the first layer of NAT using multicast but it is harder to figure out how to discover and control NATs in the next layer up.

3.1.5. NAT PMP

The NAT Port Mapping Protocol (NAT PMP) allows a computer in a private network (behind a NAT router) to automatically configure the router to allow parties outside the private network to contact it. NAT PMP runs over UDP. It essentially automates the process of port forwarding. Included in the protocol is a method for retrieving the public IP address of a NAT gateway, thus allowing a client to make this public IP address and port number known to peers that may wish to communicate with it.

Many NATs do not support PMP. In those that do support it, it has similar issues with negotiation of multilayer NATs as UPnP. Video conferencing is used extensively in enterprise networks, and NAT PMP is not generally available in enterprise-class routers.

4. Difference from RFC4582

This section details the difference from [RFC4582], the base protocol specification of BFCP, required for use over an unreliable transport. The section numbers to which differences apply are indicated in parentheses in the titles of the sub-sections below.

4.1. Overview of Operation (4)

Fourth paragraph change:

> There are two types of transaction in BFCP: client-initiated transactions and server-initiated transactions. Client-initiated transactions consist of a message from a client to the floor control server and a response from the floor control server to the client. Correspondingly, server-initiated transactions consist of a message from the floor control server to a client and the associated acknowledgement message from the client to the floor control server. Both messages can be related because they carry the same Transaction ID value in their common headers.

4.1.1. Floor Participant to Floor Control Server Interface (4.1)

Before seventh paragraph (page 9), insert:

> Figures 2 and 3 below show call flows for two sample BFCP interactions when used over reliable transport. Appendix A (Editorial Note: here-in Section 4.31) shows the same sample interactions but over an unreliable transport.

4.2. COMMON-HEADER Format (5.1)

The figure below should replace Figure 5: COMMON-HEADER format.

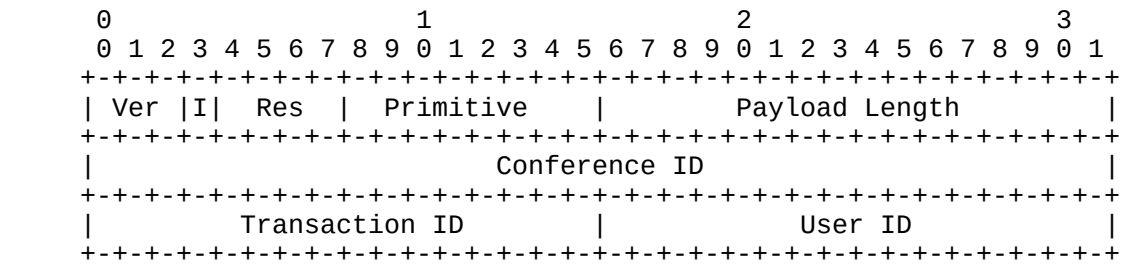


Figure 2: COMMON-HEADER format

The following text precedes "Reserved" on page 15:

> I: The Transaction Initiator (I) flag-bit has relevance only for use of BFCP over unreliable transport. When clear, it indicates that this message is a request initiating a new transaction, and the Transaction ID that follows has been generated for this transaction. When set, it indicates that this message is a response to a previous request, and the Transaction ID that follows is the one associated with that request. When BFCP is used over reliable transports, the flag has no significance and SHOULD be cleared.

The Reserved field changes name to Res due to limited space in the ASCII graphic in Figure 2. In the description of the Reserved field "the 5 bits" is changed to "the 4 bits".

The description of Transaction ID should have the final clause deleted with the reference to Section 8 remaining. The value used for server-initiated transactions MUST be non-zero when BFCP is used over unreliable transports, and this qualification shall be described in the updated Section 8.

The values below should be appended to the end of Table 1: BFCP primitives.

Value	Primitive	Direction
14	FloorRequestStatusAck	P -> S ; Ch -> S
15	ErrorAck	P -> S ; Ch -> S
16	FloorStatusAck	P -> S ; Ch -> S
17	Goodbye	P -> S ; Ch -> S ; P <- S ; Ch <- S
18	GoodbyeAck	P -> S ; Ch -> S ; P <- S ; Ch <- S

Table 1: BFCP primitives

4.3. ERROR-CODE (5.2.6)

The value below should be appended to the end of Table 5: Error Code meaning.

Value	Meaning
10	Unable to parse message
11	Use DTLS

Table 2: Error Code meaning

4.4. FloorRequestStatusAck (5.3.14)

This new subsection specifies the normative ABNF for the new primitive, FloorRequestStatusAck.

Floor participants and chairs acknowledge the receipt of a FloorRequestStatus message from the floor control server when communicating over unreliable transport. The following is the format of the FloorRequestStatusAck message:

```
FloorRequestStatusAck          =    (COMMON-HEADER)
                                   *[EXTENSION-ATTRIBUTE]
```

Figure 3: FloorRequestStatusAck format

4.5. ErrorAck (5.3.15)

This new subsection specifies the normative ABNF for the new primitive, ErrorAck.

Floor participants and chairs acknowledge the receipt of an Error message from the floor control server when communicating over unreliable transport. The following is the format of the ErrorAck message:

```
ErrorAck                    =    (COMMON-HEADER)
                                *[EXTENSION-ATTRIBUTE]
```

Figure 4: ErrorAck format

4.6. FloorStatusAck (5.3.16)

This new subsection specifies the normative ABNF for the new primitive, FloorStatusAck.

Floor participants and chairs acknowledge the receipt of a FloorStatus message from the floor control server when communicating over unreliable transport. The following is the format of the FloorStatusAck message:

```
FloorStatusAck              =    (COMMON-HEADER)
                                *[EXTENSION-ATTRIBUTE]
```

Figure 5: FloorStatusAck format

4.7. Goodbye (5.3.17)

This new subsection specifies the normative ABNF for the new primitive, Goodbye.

BFCP entities that wish to dissociate themselves from their remote participant do so through the transmission of a Goodbye. The following is the format of the Goodbye message:

```
Goodbye                     =    (COMMON-HEADER)
                                *[EXTENSION-ATTRIBUTE]
```

Figure 6: Goodbye format

4.8. GoodbyeAck (5.3.18)

This new subsection specifies the normative ABNF for the new primitive, GoodbyeAck.

BFCP entities communicating over an unreliable transport should acknowledge the receipt of a Goodbye message from a peer. The following is the format of the GoodbyeAck message:

```
GoodbyeAck                    =    (COMMON-HEADER)
                                  *[EXTENSION-ATTRIBUTE]
```

Figure 7: GoodbyeAck format

4.9. Transport (6)

An additional behavior is recommended for entities participating in communication over an unreliable transport that either wish to leave or are asked to leave an established BFCP connection, as detailed in the revised section introduction text below.

The transport over which BFCP entities exchange messages depends on how clients obtain information to contact the floor control server (e.g. using an SDP offer/answer exchange [RFC4583]). Two transports are supported: TCP, appropriate where entities can be sure that their connectivity is not impeded by NAT devices, media relays or firewalls; and UDP for those deployments where TCP may not be applicable or appropriate.

If a client wishes to end its BFCP association with a floor control server, it is RECOMMENDED that the client send a Goodbye message to dissociate itself from any allocated resources. If a floor control server wishes to end its BFCP association with a client (e.g. the Focus of the conference informs the floor control server that the client has been kicked out from the conference), it is RECOMMENDED that the floor control server send a Goodbye message towards the client.

4.9.1. Reliable Transport (6.1)

BFCP entities may elect to exchange BFCP messages using TCP connections. TCP provides an in-order reliable delivery of a stream of bytes. Consequently, message framing is implemented in the application layer. BFCP implements application-layer framing using

TLV-encoded attributes.

A client MUST NOT use more than one TCP connection to communicate with a given floor control server within a conference. Nevertheless, if the same physical box handles different clients (e.g. a floor chair and a floor participant), which are identified by different User IDs, a separate connection per client is allowed.

If a BFCP entity (a client or a floor control server) receives data that cannot be parsed, the entity MUST close the TCP connection, and the connection SHOULD be reestablished. Similarly, if a TCP connection cannot deliver a BFCP message and times out, the TCP connection SHOULD be reestablished.

The way connection reestablishment is handled depends on how the client obtains information to contact the floor control server. Once the TCP connection is reestablished, the client MAY resend those messages for which it did not get a response from the floor control server.

If a floor control server detects that the TCP connection towards one of the floor participants is lost, it is up to the local policy of the floor control server what to do with the pending floor requests of the floor participant. In any case, it is RECOMMENDED that the floor control server keep the floor requests (i.e., that it does not cancel them) while the TCP connection is reestablished.

To maintain backwards compatibility with older implementations of [RFC4583], BFCP entities MUST interpret the graceful close of their TCP connection from their associated participant as an implicit Goodbye message.

4.9.2. Unreliable Transport (6.2)

BFCP entities may elect to exchange BFCP messages using UDP datagrams. UDP is an unreliable transport where neither delivery nor order is assured. Each BFCP UDP datagram MUST contain exactly one BFCP message. In the event the size of a BFCP message exceeds the MTU size, the BFCP message will be fragmented at the IP layer. Considerations related to fragmentation are covered in Section 4.9.3. The message format for exchange of BFCP in UDP datagrams is the same as for a TCP stream above.

Clients MUST announce their presence to the floor control server by transmission of a Hello message. This Hello message MUST be responded to with a HelloAck message and only upon receipt can the client consider the floor control service as present and available.

As described in Section 8, each request sent by a floor participant or chair shall form a client transaction that expects an acknowledgement message back from the floor control server within a retransmission window. Concordantly, messages sent by the floor control server that are not transaction-completing (e.g. FloorStatus announcements as part of a FloorQuery subscription) are server-initiated transactions that require acknowledgement messages from the floor participant and chair entities to which they were sent.

If a BFCP entity receives data that cannot be parsed, the receiving participant MAY send an Error message with parameter value 10 indicating receipt of a malformed message. If the message can be parsed to the extent that it is able to discern that it was a response to an outstanding request transaction, the client MAY discard the message and await retransmission. BFCP entities receiving an Error message with value 10 SHOULD acknowledge the error and act accordingly.

Transaction ID values are non-sequential and entities are at liberty to select values at random. Entities MUST only have at most one outstanding request transaction at any one time. Implicit subscriptions, such as FloorRequest messages that have multiple responses as the floor control server processes intermediate states until Granted or Denied terminal states attained, can be characterized by a client-initiated request transaction whose acknowledgement is implied by the first FloorRequestStatus response from the floor control server. The subsequent changes in state for the request are new transactions whose Transaction ID is determined by the floor control server and whose receipt by the client participant shall be acknowledged with a FloorRequestStatusAck message. [Editorial note: would it be more straightforward to have all FloorRequestStatus messages acknowledged with a FloorRequestStatusAck message?]

By restricting entities to having at most one pending transaction open, both the out-of-order receipt of messages as well as the possibility for congestion are mitigated. Additional details regarding congestion control are provided in Section 4.9.2.1. A server-initiated request (e.g. a FloorStatus with an update from the floor control server) received by a participant before the initial FloorRequestStatus message that closes the client-initiated transaction that was instigated by the FloorRequest MUST be treated as superseding the information conveyed in any delinquent response. As the floor control server cannot send a second update to the implicit floor status subscription until the first is acknowledged, ordinality is maintained.

4.9.2.1. Congestion Control

BFCP may be characterized to generate "low data-volume" traffic, per the classification in [RFC5405]. Nevertheless is it necessary to ensure suitable and necessary congestion control mechanisms are used for BFCP over UDP. As described in previous paragraph every entity - client or server - is only allowed to send one request at a time, and await the acknowledging response. This way at most one datagram is sent per RTT given the message is not lost during transmission. In case the message is lost, the request retransmission timer T1 specified in Section 4.14 will fire and the message is retransmitted up to three times. The default initial interval is set to 500ms and the interval is doubled after each retransmission attempt, this is identical to the specification of the T1 timer in SIP as described in Section 17.1.1.2 of [RFC3261].

4.9.2.2. ICMP Error Handling

If a BFCP entity receives an ICMP port unreachable message mid-conversation, the entity SHOULD treat the conversation as closed (e.g. an implicit Goodbye message from the peer) and behave accordingly. The entity MAY attempt to re-establish the conversation afresh. The new connection will appear as a wholly new floor participant, chair or floor control server with all state previously held about that participant lost.

Note: This is because the peer entities cannot rely on IP and port tuple to uniquely identify the participant, nor would extending Hello to include an attribute that advertised what the entity previously was assigned as a User ID be acceptable due to session hijacking.

In deployments where NAT appliances, firewalls or other such devices are present and affecting port reachability for each entity, one possibility is to utilize the peer connectivity checks, relay use and NAT pinhole maintenance mechanisms defined in ICE [RFC5245].

4.9.3. Large Message Considerations

Large messages become a concern when using BFCP if the overall size of a single BFCP message exceeds that representable within the 16-bit Payload Length field of the COMMON-HEADER. When using UDP, there is the added concern that a single BFCP message can be fragmented at the IP layer if its overall size exceeds the MTU threshold of the network.

The target use cases for BFCP via UDP typically involve relatively small BFCP messages. Combining that with the goal of minimizing differences to the standard BFCP specification, BFCP entities SHOULD

ensure that their messages are smaller than the recommended MTU size of 1300 bytes when encoded to minimize the likelihood of fragmentation in route to their peer entity.

Note: While outside the scope of this document, the definition of additional mechanisms to further address BFCP message fragmentation are welcome. Potential mechanisms mentioned previously include:

- a mechanism for splitting a single large message into additive messages. The mechanism defined for RELOAD in section 5.7 of [I-D.ietf-p2psip-base] has been identified as a good candidate.

- an applicability statement on those BFCP messages and/or attributes deemed as inappropriate for use over transports where fragmentation is a concern.

- a SIP event package to deliver information to the endpoints.

4.10. Lower-Layer Security (7)

Expand the section to mandate support for DTLS when transport over UDP is used such that it reads as follows:

BFCP relies on lower-layer security mechanisms to provide replay and integrity protection and confidentiality. BFCP floor control servers and clients (which include both floor participants and floor chairs) MUST support TLS for transport over TCP and MUST support DTLS for transport over UDP [RFC5246]. Any BFCP entity MAY support other security mechanisms.

BFCP entities MUST support, at a minimum, the TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite [RFC5246].

Which party, the client or the floor control server, acts as the TLS/DTLS server depends on how the underlying TCP/DTLS connection is established. For example, when the TCP/DTLS connection is established using an SDP offer/answer exchange [RFC4583], the answerer (which may be the client or the floor control server) always acts as the TLS/DTLS server.

4.11. Protocol Transactions (8)

The final clause of the introduction to section 8 should be read as:

> Since they do not trigger any response, their Transaction ID is set to 0 when used over reliable transports, but must be non-zero and unique in the context of outstanding transactions over unreliable transports.
>
> When using BFCP over unreliable transports, all requests will use retransmit timer T1 (see Section 4.13) until the transaction is completed.

4.12. Server Behavior (8.2)

The final clause of this section should be read as:

> Server-initiated transactions MUST contain a Transaction ID equal to 0 when BFCP is used over reliable transports. Over unreliable transport, the Transaction ID shall have the same properties as for client-initiated transactions: the server MUST set the Transaction ID value in the common header to a number that is different from 0 and that MUST NOT be reused in another message from the server until the appropriate response from the client is received for the transaction. The server uses the Transaction ID value to match this message with the response from the floor participant or floor chair.

4.13. Timers (8.3)

New section:

> When BFCP entities are communicating over an unreliable transport, two retransmission timers are employed to help mitigate against loss of datagrams. Retransmission and response caching are not required when BFCP entities communicate over reliable transports.

4.14. Request Retransmission Timer, T1 (8.3.1)

T1 is a timer that schedules retransmission of a request until an appropriate response is received or until the maximum number of retransmissions have occurred. The timer doubles on each re-transmit, failing after three unacknowledged transmission attempts.

If a valid response is not received for a client- or server-initiated transaction, the implementation MUST consider the BFCP association as failed. Implementations SHOULD follow the reestablishment procedure described in section 6 (e.g. initiate a new offer/answer [RFC3264]

exchange). Alternatively, they MAY continue without BFCP and therefore not be participant in any floor control actions.

4.15. Response Retransmission Timer, T2 (8.3.2)

T2 is a timer that, when fires, signals that the BFCP entity can release knowledge of the transaction against which it is running. It is started upon the first transmission of the response to a request and is the only mechanism by which that response is released by the BFCP entity. Any subsequent retransmissions of the same request can be responded to by replaying the cached response, whilst that value is retained until the timer has fired.

T2 shall be set such that it encompasses all legal retransmissions per T1 plus a factor to accommodate network latency between BFCP entities.

4.16. Timer Values (8.3.3)

The table below defines the different timers required when BFCP entities communicate over an unreliable transport.

Timer	Description	Value/s
T1	Initial request retransmission timer	0.5s
T2	Response retransmission timer	10s

Table 3: Timers

The default value for T1 is 500 ms, this is an estimate of the RTT for completing the transaction. T1 MAY be chosen larger, and this is RECOMMENDED if it is known in advance that the RTT is larger. Regardless of the value of T1, the exponential backoffs on retransmissions described in Section 4.14 MUST be used.

4.17. Authentication and Authorization (9)

The first sentence of the second paragraph should be read as:

> BFCP supports TLS/DTLS mutual authentication between client and floor control servers, as specified in section 9.1.

4.17.1. TLS Based Mutual Authentication (9.1)

Change each instance of "TLS" to "TLS/DTLS", and each instance of "TCP" to "TCP/UDP".

4.18. Receiving a Response [to a FloorRequest Message] (10.1.2)

Prepend the sentence below at the start of this subsection:

> When communicating over unreliable transport and upon receiving a FloorRequest from a participant, the floor control server MUST respond with a FloorRequestStatus message within the transaction failure window to complete the transaction.

4.19. Receiving a Response [to a FloorRelease Message] (10.2.2)

Prepend the sentence below at the start of this subsection:

> When communicating over unreliable transport and upon receiving a FloorRelease from a participant, the floor control server MUST respond with a FloorRequestStatus message within the transaction failure window to complete the transaction.

4.20. Receiving a Response [to a ChairAction Message] (11.2)

Prepend the sentence below at the start of this subsection:

> When communicating over unreliable transport and upon receiving a ChairAction from a participant, the floor control server MUST respond with a ChairActionAck message within the transaction failure window to complete the transaction.

4.21. Receiving a Response [to a FloorQuery Message] (12.1.2)

Prepend the sentence below at the start of this subsection:

> When communicating over unreliable transport and upon receiving a FloorQuery from a participant, the floor control server MUST respond with a FloorStatus message within the transaction failure window to complete the transaction.

4.22. Receiving a Response [to a FloorRequestQuery Message] (12.2.2)

Prepend the sentence below at the start of this subsection:

> When communicating over unreliable transport and upon receiving a FloorRequestQuery from a participant, the floor control server MUST respond with a FloorRequestStatus message within the

transaction failure window to complete the transaction.

4.23. Receiving a Response [to a UserQuery Message] (12.3.2)

Prepend the sentence below at the start of this subsection:

> When communicating over unreliable transport and upon receiving a UserQuery from a participant, the floor control server MUST respond with a UserStatus message within the transaction failure window to complete the transaction.

4.24. Receiving a Response [to a Hello Message] (12.4.2)

Prepend the sentence below at the start of this subsection:

> When communicating over unreliable transport and upon receiving a Hello from a participant, the floor control server MUST respond with a HelloAck message within the transaction failure window to complete the transaction.

4.25. Reception of a FloorRequestStatus Message (13.1.3)

The sentence below shall appear as a new subsection:

> When communicating over unreliable transport and upon receiving a FloorRequestStatus message from a floor control server, the participant MUST respond with a FloorRequestStatusAck message within the transaction failure window to complete the transaction.

4.26. Reception of a FloorStatus Message (13.5.3)

The sentence below shall appear as a new subsection:

> When communicating over unreliable transport and upon receiving a FloorStatus message from a floor control server, the participant MUST respond with a FloorStatusAck message within the transaction failure window to complete the transaction.

4.27. Reception of an Error Message (13.8.1)

The sentence below shall appear as a new subsection:

> When communicating over unreliable transport and upon receiving an Error message from a floor control server, the participant MUST respond with a ErrorAck message within the transaction failure window to complete the transaction.

4.28. Security Considerations (14)

Change each instance of "TLS" to "TLS/DTLS", and each instance of "TCP" to "TCP/UDP".

4.29. IANA Considerations - Primitive Subregistry (15.2)

This section instructs the IANA to register the following new values for the BFCP primitive subregistry.

Value	Primitive	Reference
14	FloorRequestStatusAck	RFC 4582bis
15	ErrorAck	RFC 4582bis
16	FloorStatusAck	RFC 4582bis
17	Goodbye	RFC 4582bis
18	GoodbyeAck	RFC 4582bis

Table 4: BFCP primitive subregistry

4.30. IANA Considerations - Error Code Subregistry (15.4)

This section instructs the IANA to register the following new values for the BFCP Error Code subregistry.

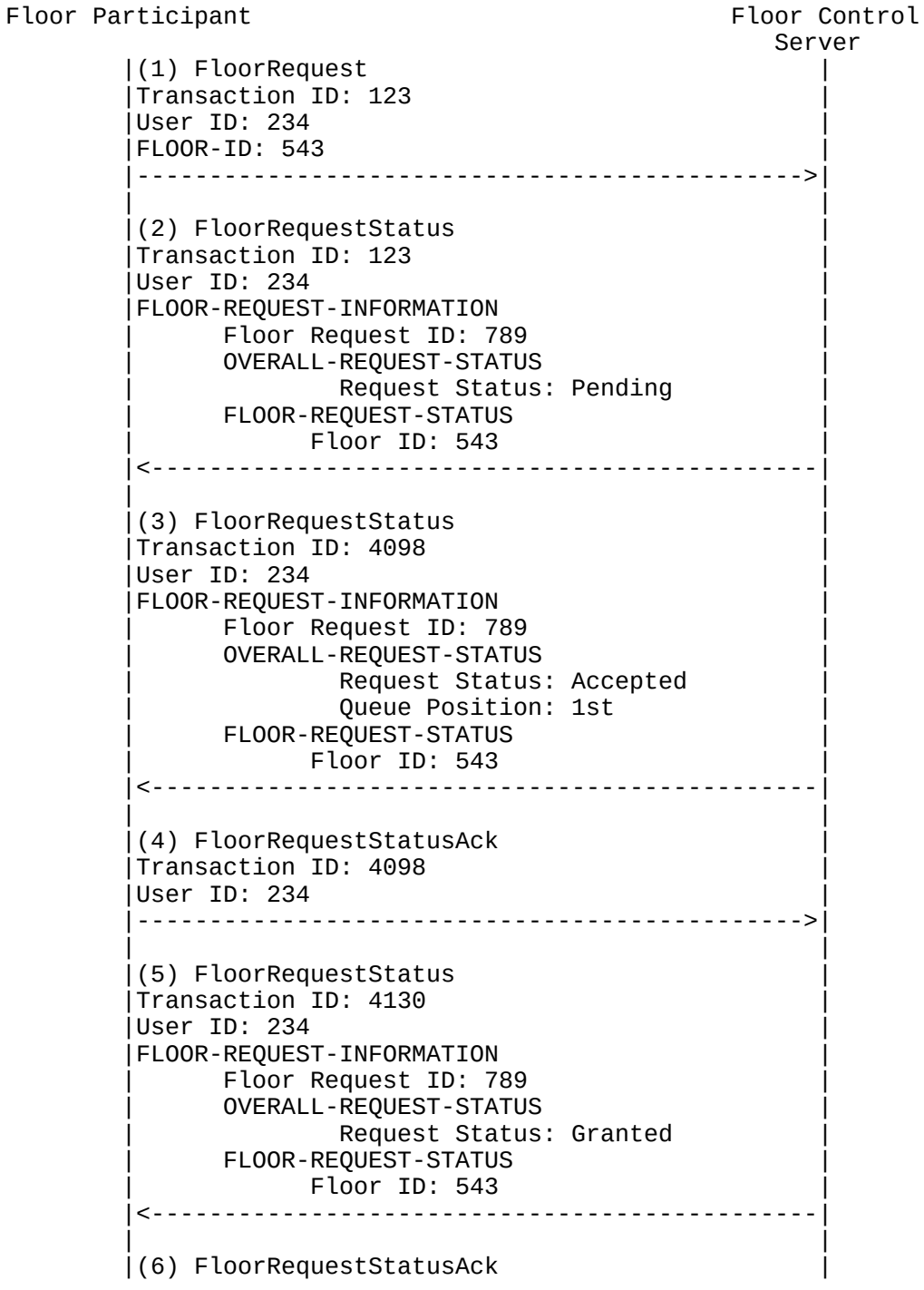
Value	Meaning	Reference
10	Unable to parse message	RFC 4582bis
11	Use DTLS	RFC 4582bis

Table 5: BFCP Error Code subregistry

4.31. Example Call Flows for BFCP over Unreliable Transport (Appendix A)

With reference to Section 4.1, the following figures show representative call-flows for requesting and releasing a floor, and obtaining status information about a floor when BFCP is deployed over an unreliable transport. The figures here show a loss-less interaction.

Editorial Note: A future version of this draft will show an example with lost packets due to unreliable transport, as well as examples on usage of DTLS and STUN in call the setup phase.


Floor Participant
Floor Control Server
(1) FloorRequest
Transaction ID: 123
User ID: 234
FLOOR-ID: 543
(2) FloorRequestStatus
Transaction ID: 123
User ID: 234
FLOOR-REQUEST-INFORMATION
Floor Request ID: 789
OVERALL-REQUEST-STATUS
Request Status: Pending
FLOOR-REQUEST-STATUS
Floor ID: 543
(3) FloorRequestStatus
Transaction ID: 4098
User ID: 234
FLOOR-REQUEST-INFORMATION
Floor Request ID: 789
OVERALL-REQUEST-STATUS
Request Status: Accepted
Queue Position: 1st
FLOOR-REQUEST-STATUS
Floor ID: 543
(4) FloorRequestStatusAck
Transaction ID: 4098
User ID: 234
(5) FloorRequestStatus
Transaction ID: 4130
User ID: 234
FLOOR-REQUEST-INFORMATION
Floor Request ID: 789
OVERALL-REQUEST-STATUS
Request Status: Granted
FLOOR-REQUEST-STATUS
Floor ID: 543
(6) FloorRequestStatusAck

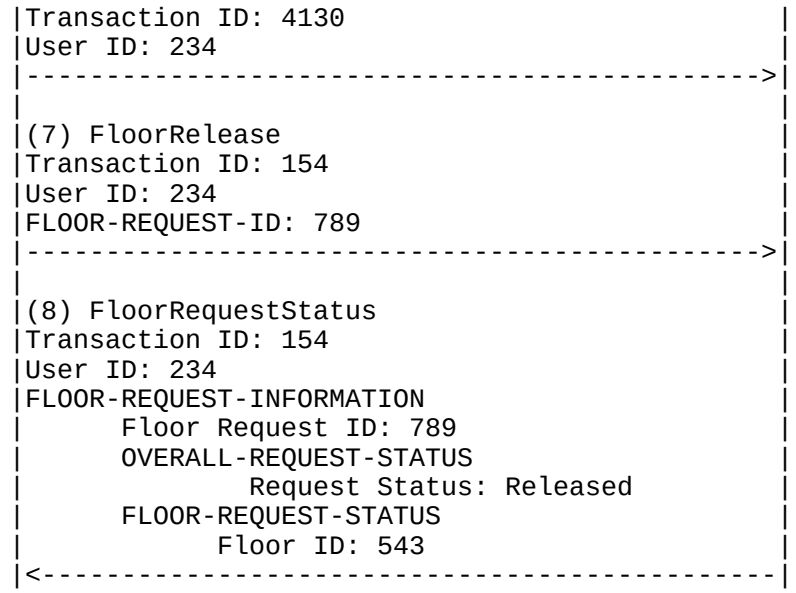
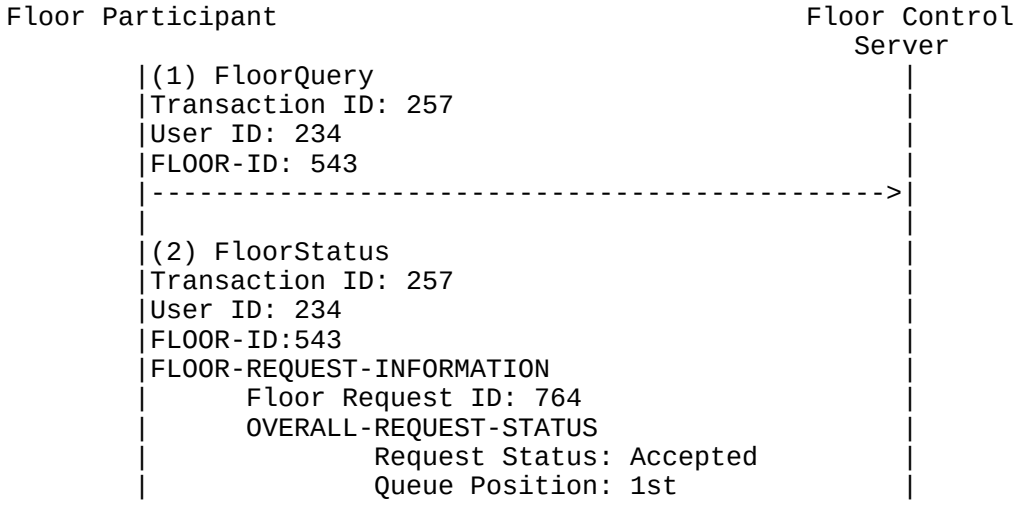
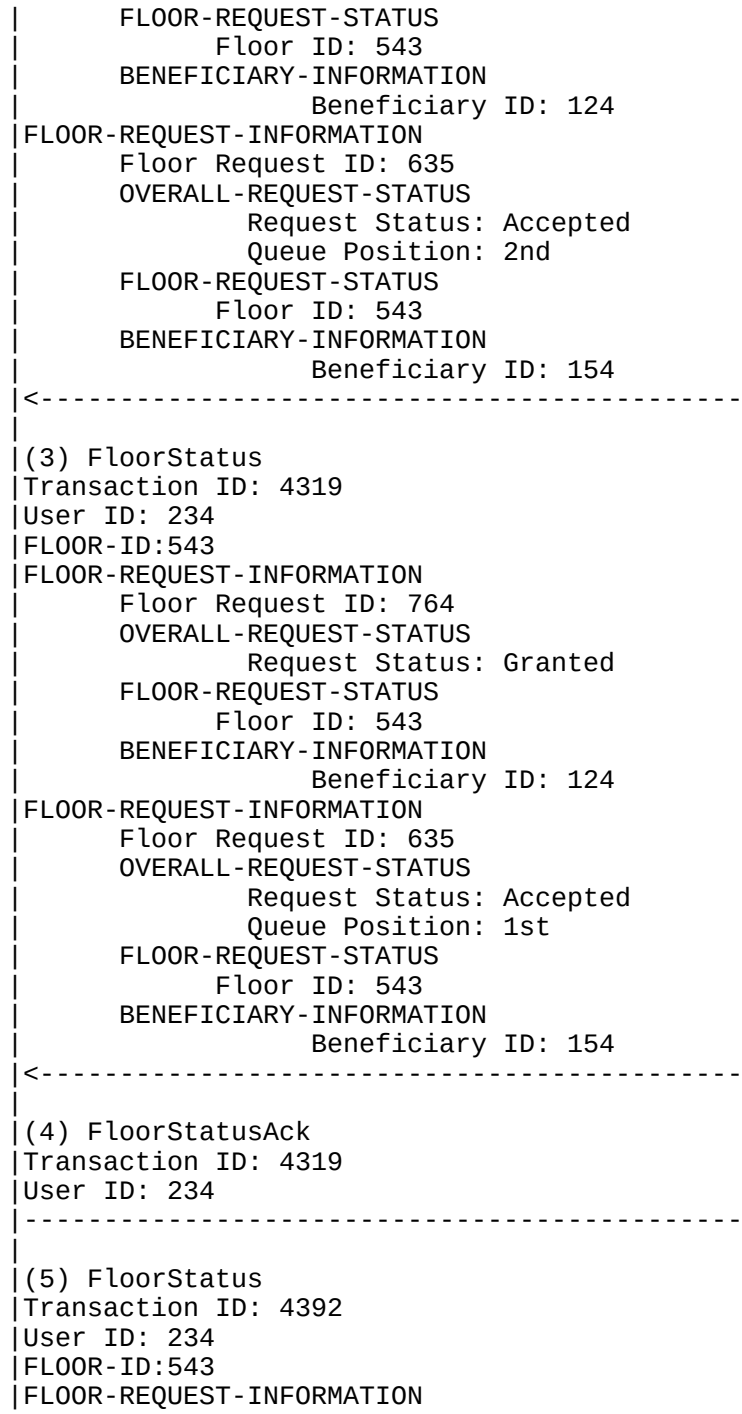


Figure 8: Requesting and releasing a floor

Note that in Figure 8, the FloorRequestStatus message from the floor control server to the floor participant is a transaction-closing message as a response to the client-initiated transaction with Transaction ID 154. It does not and SHOULD NOT be followed by a FloorRequestStatusAck message from the floor participant to the floor control server.




FLOOR-REQUEST-STATUS
Floor ID: 543
BENEFICIARY-INFORMATION
Beneficiary ID: 124
FLOOR-REQUEST-INFORMATION
Floor Request ID: 635
OVERALL-REQUEST-STATUS
Request Status: Accepted
Queue Position: 2nd
FLOOR-REQUEST-STATUS
Floor ID: 543
BENEFICIARY-INFORMATION
Beneficiary ID: 154
(3) FloorStatus
Transaction ID: 4319
User ID: 234
FLOOR-ID:543
FLOOR-REQUEST-INFORMATION
Floor Request ID: 764
OVERALL-REQUEST-STATUS
Request Status: Granted
FLOOR-REQUEST-STATUS
Floor ID: 543
BENEFICIARY-INFORMATION
Beneficiary ID: 124
FLOOR-REQUEST-INFORMATION
Floor Request ID: 635
OVERALL-REQUEST-STATUS
Request Status: Accepted
Queue Position: 1st
FLOOR-REQUEST-STATUS
Floor ID: 543
BENEFICIARY-INFORMATION
Beneficiary ID: 154
(4) FloorStatusAck
Transaction ID: 4319
User ID: 234
(5) FloorStatus
Transaction ID: 4392
User ID: 234
FLOOR-ID:543
FLOOR-REQUEST-INFORMATION

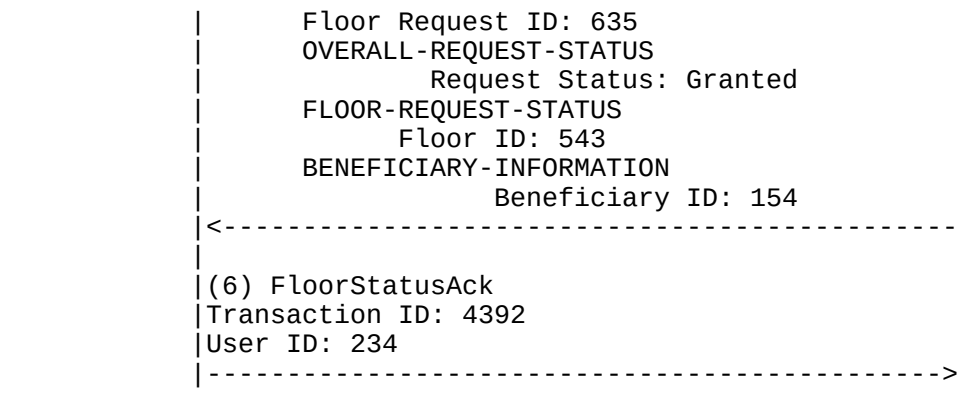


Figure 9: Obtaining status information about a floor

5. Revision of RFC4583

This section details revisions to [RFC4583], the SDP format for specifying BFCP streams. The section number to which updates apply are indicated in parentheses in the titles of the sub-sections below.

5.1. Fields in the 'm' Line (3)

The section shall be re-written to remove reference to the exclusivity of TCP as a transport for BFCP streams.

1. In paragraph four, "... will initiate its TCP connection ..." becomes "... will direct BFCP messages ..."

2. In paragraph four, delete "Since BFCP only runs on top of TCP, the port is always a TCP port."

3. Change paragraph five, "We define two new values ... ", to, "We define four new values for the transport field: TCP/BFCP, TCP/TLS/BFCP, UDP/BFCP, and UDP/TLS/BFCP. TCP/BFCP is used when BFCP runs directly on top of TCP, and TCP/TLS/BFCP is used when BFCP runs on top of TLS, which in turn runs on top of TCP. Similarly, UDP/BFCP is used when BFCP runs directly on top of UDP, and UDP/TLS/BFCP is used when BFCP runs on top of DTLS [RFC4347], which in turn runs on top of UDP.

5.2. Authentication (8)

In last paragraph, change "When TLS is used, once the underlaying TCP connection is established" to "When TLS is used with TCP, once the underlying connection is established".

5.3. Security Considerations (10)

Append to the first paragraph, "Furthermore, when using DTLS over UDP, considerations for its use with RTP and RTCP are presented in [RFC5763]. The requirements for the offer/answer exchange, as listed in Section 5 of that document, MUST be followed."

5.4. Registration of SDP 'proto' Values (11.1)

This section should be renamed now that there are more values to register in the SDP parameters registry, with the following added to the table:

Value	Reference
UDP/BFCP	RFC 4583bis
UDP/TLS/BFCP	RFC 4583bis

Table 6: Value for the SDP 'proto' field

6. NAT Traversal

One of the key benefits when using UDP for BFCP communication is the ability to leverage the existing NAT traversal infrastructure and strategies deployed to facilitate transport of the media associated with the video conferencing sessions. Depending on the given deployment, this infrastructure typically includes some subset of ICE [RFC5245].

In order to facilitate the initial establishment of NAT bindings, and to maintain those bindings once established, BFCP over UDP entities are RECOMMENDED to use STUN [RFC5389] for keep-alives, as described for SIP [RFC5626]. This results in each BFCP entity sending a packet, both to open the pinhole and to learn what IP/port the NAT assigned for the binding.

In order to facilitate traversal of BFCP packets through NATs, BFCP over UDP entities are RECOMMENDED to use symmetric ports for sending and receiving BFCP packets, as recommended for RTP/RTCP [RFC4961].

7. Future Work

This draft reflects a work in progress, with at least the following items to be documented and/or revised:

Example signaling flows: A later version of this draft will include further examples of signaling exchanges over unreliable transport as a visual aid and reference for implementers, including updated transactions, message retransmission, usage of DTLS during call setup, and combined usage of DTLS and STUN.

IANA Considerations: The current considerations are based on this extension being standards track. Now that it is informational, it can no longer add attributes to the registries defined for BFCP per the requirements stated in [RFC4582]. As such, the sections related to IANA considerations will need to be reworked.

8. Acknowledgements

We acknowledge substantial contributions to one or more previous versions of this draft from Trond G. Andersen, Alfred E. Heggestad, Gonzalo Camarillo, Roni Even, Lorenzo Miniero, Joerg Ott, Hadriel Kaplan, Dan Wing, Cullen Jennings, David Benham, and Alan Ford.

9. References

9.1. Normative References

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

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

[RFC4347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", RFC 4347, April 2006.

[RFC4582] Camarillo, G., Ott, J., and K. Drage, "The Binary Floor Control Protocol (BFCP)", RFC 4582, November 2006.

[RFC4583] Camarillo, G., "Session Description Protocol (SDP) Format for Binary Floor Control Protocol (BFCP) Streams", RFC 4583, November 2006.

[RFC4961] Wing, D., "Symmetric RTP / RTP Control Protocol (RTCP)", BCP 131, RFC 4961, July 2007.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.

[RFC5626] Jennings, C., Mahy, R., and F. Audet, "Managing Client-Initiated Connections in the Session Initiation Protocol (SIP)", RFC 5626, October 2009.

9.2. Informative References

[I-D.ietf-mmusic-ice-tcp]
Rosenberg, J., Keranen, A., Lowekamp, B., and A. Roach, "TCP Candidates with Interactive Connectivity Establishment (ICE)", draft-ietf-mmusic-ice-tcp-15 (work in progress), September 2011.

[I-D.ietf-mmusic-media-path-middleboxes]
Stucker, B. and H. Tschofenig, "Analysis of Middlebox Interactions for Signaling Protocol Communication along the Media Path", draft-ietf-mmusic-media-path-middleboxes-03 (work in progress), July 2010.

[I-D.ietf-p2psip-base]
Jennings, C., Lowekamp, B., Rescorla, E., Baset, S., and H. Schulzrinne, "REsource LOcation And Discovery (RELOAD) Base Protocol", draft-ietf-p2psip-base-19 (work in progress), October 2011.

[I-D.manner-tsvwg-gut]
Manner, J., Varis, N., and B. Briscoe, "Generic UDP Tunnelling (GUT)", draft-manner-tsvwg-gut-02 (work in progress), July 2010.

[IMC05] Guha, S. and P. Francis, "Characterization and Measurement of TCP Traversal through NATs and Firewalls", 2005, <http://saikat.guha.cc/pub/imc05-tcpnat.pdf/>.

[P2PNAT] Ford, B., Srisuresh, P., and D. Kegel, "Peer-to-Peer Communication Across Network Address Translators", April 2005, <http://www.brynosaurus.com/pub/net/p2pnat.pdf/>.

[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.

[RFC4380] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, February 2006.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.

[RFC5405] Eggert, L. and G. Fairhurst, "Unicast UDP Usage Guidelines for Application Designers", BCP 145, RFC 5405, November 2008.

[RFC5763] Fischl, J., Tschofenig, H., and E. Rescorla, "Framework for Establishing a Secure Real-time Transport Protocol (SRTP) Security Context Using Datagram Transport Layer Security (DTLS)", RFC 5763, May 2010.

[RFC6081] Thaler, D., "Teredo Extensions", RFC 6081, January 2011.

Appendix A. Change History

A.1. draft-sandbakken-dispatch-bfcp-udp-02 to -03

1. Added fragmentation and reassembly mechanism defined for RELOAD as a candidate mechanism for consideration for BFCP when transported over UDP.

2. Added ERROR-CODE to indicate DTLS is required.

3. Added UDP/TLS/BFCP as 4th transport value for BFCP.

4. Added requirement to follow offer/answer procedure in [RFC5763] when using DTLS over UDP for BFCP.

A.2. draft-sandbakken-dispatch-bfcp-udp-01 to -02

1. Switched from standards track to informational.

2. Added section on motivation, including alternatives considered, to address issues raised at IETF 79 and on various workgroup aliases.

3. Changed semantics of the Transaction Initiator (I) flag-bit.

4. Expanded transport section to more explicitly call out considerations regarding congestion control and ICMP errors, and

add considerations for large messages.

5. Updated security related sections and added authentication section to address DTLS when using UDP.

6. Added section on NAT Traversal.

7. Some editorial changes.

A.3. draft-sandbakken-dispatch-bfcp-udp-00 to -01

1. Decision made to not increase the protocol version number as a result of this extension. Certain aspects of this draft require different behaviors depending on whether a reliable or unreliable transport is being used, e.g. server-initiated transactions having Transaction ID 0 over reliable transports without acknowledgements versus non-zero and active-unique with an acknowledgement message when entities communicate over unreliable transports. As the graceful-close behavior of [RFC4582] is still allowed for TCP-based implementations without mandating the use of the new Goodbye message, there is no need to change the version number.

2. Removed the - a bit too verbose - rationale/motivation text describing background and why other approaches where not chosen. Was OK for a -00 draft, not strictly needed.

3. Not mandate ICE as a SHALL, but leave it as a non-mandatory way of solving the potential need for NAT/FW traversal.

4. Emphasized that the reference to DTLS-SRTP are merely informational.

5. A dash of polish and nitpicking added, some typos fixed.

A.4. draft-sandbakken-xcon-bfcp-udp-02 to draft-sandbakken-dispatch-bfcp-udp-00

1. Draft name change. As XCON WG is closing this draft is submitted to Dispatch WG as the arena of discussion.

2. Moved Transaction Identifier bit (I) from the Transaction ID to one of the current 5 reserved bits. Keep current Transaction ID syntax and semantics. Avoid potential problems with existing TCP based implementations.

3. The way congestion control is taken care of is explained, with reference to [RFC5405]. One message per RTT. Backoff and

normative behavior for timer T1 clarified.

4. Mandated support for DTLS in case unreliable transport (i.e. UDP) is implemented. Details and examples to be included. Model after [RFC5763], details on how to adapt the SRTP associated details to BFCP and whether a reference or copying the text across and changing is needed.

5. Added the Rationale and Scope section to position and explain the motivation for this draft more in detail.

6. A number of typos and editorial changes.

A.5. draft-sandbakken-xcon-bfcp-udp-01 to -02

1. Stepped away from changing semantics and directionality of Hello and HelloAck messages for pinhole establishment and keep-alive in favor of ICE toolset, particularly as this would have not resolved connectivity establishment as a precursor to deployment of DTLS [RFC4347] as a transport security mechanism.

2. Change to COMMON-HEADER to reserve bit-16 of Transaction ID to show originator of transaction such that request/response and response/acknowledgement mapping can be maintained without colliding randomly chosen Transaction IDs. This also avoids a three-way handshake scenario around FloorRequest where the implicit acknowledgement (in FloorRequestStatus) might also be interpreted as a transaction opening request on the part of the floor control server.

3. Defined additional timer (T2) to soak up lost responses without additional processing.

4. Restricted outstanding transactions to only one in-flight per direction at any one time to mitigate re-ordering issues.

5. Defined entity behavior when transactions timeout.

6. Specified initial suggestion for how to minimize fragmentation of messages.

7. Removed consideration of TCP-over-UDP after internal review.

8. Re-stated DTLS as likely preferred mechanism of securing transport, although this investigation is on-going.

A.6. draft-sandbakken-xcon-bfcp-udp-00 to -01

1. Refactored to a format that represents explicit changes to base RFCs.

2. Introduction of issues currently under investigation that preclude adoption.

3. Specified retransmission timer for requests.

Authors' Addresses

Charles Eckel
Cisco
170 West Tasman Drive
San Jose, CA 95134
United States

Email: eckelcu@cisco.com

Tom Kristensen
Cisco
Philip Pedersens vei 22
N-1366 Lysaker
Norway

Email: tomkrist@cisco.com, tomkri@ifi.uio.no

Mark K. Thompson
Cisco
Ruscombe Business Park
Ruscombe, England
UK

Email: markth2@cisco.com

Geir A. Sandbakken
Cisco
Philip Pedersens vei 22
N-1366 Lysaker
Norway

Email: geirsand@cisco.com

Eoin McLeod
Cisco
Ruscombe Business Park
Ruscombe, England
UK

Email: eoimcleo@cisco.com