Best Practices for Role-Based Video Streams (RBVS) in SIP

IMTC SIP Parity Group

Version 33

July 13, 2011

Table of Contents

1. Overview

This document describes the best practices for implementing role-based video stream (RBVS) functionality in SIP which is comparable to H.239 for H.323 systems. In particular, it covers the use of video for content sharing, including floor control, alongside main video. H.239 is the ITU standard for implementing role management of multiple streams in H.323 and H.320. It specifies mechanisms for:

1. Adding additional media channels to conferences,
2. Designating roles of video media channels ("Live" or "Presentation") and
3. Controlling the presenter of the "Presentation" video stream during a conference.

Under H.239, video streams can be given roles. A role is used to designate the purpose of each video stream, which determines how the streams should be presented and processed. A video stream without a role is called the 'main' video channel. A video stream with role 'Live' is handled as a normal, bidirectional second video stream. A video stream with role 'Presentation' is handled as a presentation to be distributed to all endpoints. H.239 includes messaging which makes it possible for users to designate and hand-off the role of presenter during the meeting through the use of tokens.

In the H.239 architecture, roles are indicated by "role labels" on a channel and specify both the purpose of the stream carried on a channel, and how the stream should be presented at the end-user system and processed by an MCU. A role label may be assigned to any channel (audio, video, or data) where it is useful to establish policies for presentation, management, or distribution of the information on that channel. However, the current version of H.239 only defines roles for a second video channel.

Through H.239, video conferencing systems are able to make intelligent decisions about where and when, to display "live" streams and "presentation" streams on their available monitors. For example, in a single monitor system, a video conferencing system will typically display the "presentation" video on the monitor and the far-end, "main" video in a Picture-in-Picture (PiP) or not display it at all. In a dual monitor system, the video conferencing system will typically display "presentation" video on the main monitor and the far-end "main" video on the secondary monitor.

While the functionality described in this document details the best practices for implementing role-based video stream (RBVS) functionality in SIP, it does not exclude the use of multiple media streams in other manners. For example, RFC 3264 describes the manner in which SIP user agents should treat the case where multiple media streams of the same type appear in the offer. It says that "if multiple media streams of the same type are present in an offer, it means that the offerer wishes to send (and/or receive) multiple streams of that type at the same time." In this case, streams of the same media type do not have any specific semantics.

'This "best practices" document does not preclude the use of other functionalities involving multiple streams, such as sending re-INVITES to add additional video streams during a session, or escalating an audio-only session to a session with multiple video streams.

In order to achieve H.239-type mechanisms in SIP, the following functionality needs to be implemented:

1. Methods to designate the role of each media channel, so that it is possible to designate the "main" video channel and "Live" or "Presentation" for the second video channel.

2. Methods to control which participant in the conference is the "presenter" for each "Presentation" channel.
3. Methods to negotiate the use of role-based video stream (RBVS).

This document describes the best practices in SIP to achieve these functionalities. It also describes the relationship between the use of multiple streams in role-based video and the use of multiple streams without specific semantics describing their roles. Other uses of multiple steams are possible, but are outside the scope of this document.

Security considerations for control and media channels utilized for this functionality are outside the scope of this document and will be addressed in the IMTC SIP Parity Group's "Security Best Practices" document.

2. Role-Based Video Stream (RBVS) "Best Practices" Profile

The "Best Practices" Profile defined in this document describes how SIP-based video endpoints are required to behave in order to comply with RBVS. Within this document, the term endpoint should be interpreted as meaning a device that acts as a SIP user agent. This includes, but is not limited to, a single user endpoint, an MCU, or a Back-to-Back User Agent (B2BUA). Table 2.1 provides a comparison between this profile and the corresponding mechanisms used in H323. An alternate SIP-based approach in which BFCP is transported over TCP, instead of over UDP, has been found to not work well with commonly deployed firewall/NAT traversal solutions. However, as implementations of this approach exist, the option of falling back to TCP-based BFCP is included in Table 2.1 and described in Appendix B as reference.

Function	H.239	"Best Practices" Profile
Designating Channel Roles (section 3)	h239ExtendedVideoCapability roleLabel	RFC 4796 content attribute
Token Control Messages (section 4.1)	H.239 Control & Indication messages	BFCP
Token Control Channel (section 4.2)	H.245	UDP-based BFCP
Offer/Answer Exchange (section 5)	H.245	Offer UDP-based BFCP as indicator of support of RBVS. Send re-INVITE for TCP-based BFCP if far-end doesn't support UDP-based BFCP (optional)

Table 2.1 Mechanisms used in Role-Based Video Stream Implementations

3. Designating the Roles of the Media Channels

3.1 Approach Used in H.239 for Designating Channel Roles

This section contains informative information only. It is included to provide background information regarding which video channel roles can be designated in H.239.

In H.239, an endpoint can signal the capabilities of its main video channel and a secondary video channel. The capabilities include the role that the secondary channel will play. The defined roles for the second channel are:

- ***Live*** – video is processed normally; suitable for live video of people.
- ***Presentation*** – a token-managed presentation to be distributed to all devices.

The "Live" role indicates that the video channel shall be distributed, managed, and presented using traditional means. The "Live" role is appropriate for live video of meeting participants. The "Live" video channel supplements the "main" video channel. It should carry a stream that is less important to display at end-user systems than the "main" channel. "Live" video is two-way transmission. Multiple devices may transmit "Live" video simultaneously.

The "Presentation" role is used to indicate that the video channel contains a presentation (also known as content) that is intended to be seen by all conference participants. Transmission on the Presentation channel is managed by the token mechanism in order to provide one-way transmission. There is only one token in a conference.
H.239 is not used to indicate the role of audio streams. Any audio associated with the video channels is generally mixed into a single audio channel.

3.2 Best Practices in SIP for Designating Roles for Media Channels

This section describes the methods SIP-based video endpoints are required to utilize to conform to the RBVS "Best Practices" Profile. SIP-based video endpoints MUST use the media-level value attribute, 'content' defined in RFC 4796 to designate the role of video channels. The purpose of the 'content' attribute is to enable an application to make a decision on how to display each video stream and to enable video streams to be mapped into the main, presentation and live roles defined in H.239. The use of the 'content' attribute for audio streams is out of scope.

RFC 4796 specifies five pre-defined values for the 'content' attribute. Other values can be defined in the future. The pre-defined values are:

- ***slides:*** This designates a media stream that includes presentation slides. The media type can be, for example, a video stream or a set of instant messages with pictures. Typical use cases for this type of media stream are online seminars and courses. This is similar to the "presentation" role in H.239.
- ***speaker:*** This designates an image from the speaker. The media could be a video stream or a still image. Typical uses for this type of media stream are online seminars and courses.
- ***sl:*** This designates a media stream contains sign language. The media type is a video stream. Typical use for this media type is where the audio stream is translated into sign language.

- ***main:*** For video, this designates that the main stream taken from the main source. A typical use case for this type of media stream is a concert, where the camera is shooting the performer. This is similar to the "main" video channel in H.239.
- ***alt:*** For video, this means that the main stream is taken from the alternative source. Typical use case for this type of media stream is the video of an entire conference room while the main media stream is the video of the speaker. This is similar to the "live" role in H.239.

The following is an example of the SDP session description that uses the 'content' attribute:

```
v=0
o=Alice 292742730 29277831 IN IP4 131.163.72.4
s=lecture
c=IN IP4 131.164.74.2
t=0 0
m=video 52886 RTP/AVP 31
a=rtpmap:31 H261/90000
a=content:slides
m=video 53334 RTP/AVP 31
a=rtpmap:31 H261/90000
a=content:main
```

For SIP-based RBVS implementations, endpoints MUST use the following content attributes:

1. The "slides" content attribute MUST be used to assign the H.239 "presentation" role to a video channel.
2. The "alt" content attribute MUST be used to assign the H.239 "live" role to a video channel.
3. The "main" content attribute MUST be used to designate the main video channel.

Unlike H.239 where there are only two channel roles, "live" and "presentation", in SIP, there are more roles, and it is possible that endpoints and MCUs will have to deal with the situation where all of the participants in a call do not designate the same roles for their channels. For example, one endpoint in a multipoint conference may use the "speaker" and "slides" roles for its video channels, while another may use "main" and "alt" for its video channels. Since there are no mechanisms to enable MCUs and endpoints to interoperate in this type of situation, all SIP-based endpoints conforming to the RBVS "Best Practices" Profile MUST use the attributes as specified in this section. The handling of other 'content' attribute values is considered out of scope.

4. Management and Control of the Presentation Token

4.1 Protocol for Token Control

4.1.1 Approach Used in H.239 for Controlling the Presenter Role

This section contains informative information only. It is included to provide background information regarding how the presenter role is controlled in H.239.

The H.239 "Presentation" channel is controlled by a token. There is one token in a conference. The holder of the token is the presenter and all of the endpoints in the conference typically display the

presenter's stream on their content rendering display. The endpoints use a set of Control & Indication messages to pass the token from one endpoint to another. In a centralized, multipoint conference, the MCU holds and manages the token.

Control & Indication Messages: In H.239 there are two kinds of control messages. The first set of messages control token passing. There are four of these messages:

- ***presentationTokenRequest:*** a request by the sender to acquire the indicated token,
- ***presentationTokenResponse:*** the device receiving the request responds with this message either accepting or rejecting the request,
- ***presentationTokenRelease:*** sent by a device holding the token in order to relinquish the token;
- ***presentationTokenIndicateOwner:*** sent periodically by the device holding the token, and forwarded by MCUs and gateways.

To send a presentation video stream, devices that support roles request the token. When the device obtains the token, it opens the channel (if closed); indicates video active, and starts sending the stream.

The other type of control message is for flow control which is outside the scope of this document.

4.1.2 Best Practices in SIP for Controlling the Presenter Role

This section describes the methods SIP-based video endpoints are required to utilize to conform to the RBVS "Best Practices" Profile. SIP-based video endpoints MUST implement token management and control using the Binary Floor Control Protocol (BFCP), which is specified in RFC 4582. Obtaining the token in H.239 is equivalent to obtaining the floor using BFCP in SIP. Appendix A provides brief, informational background about BFCP. SIP-based video endpoints conforming to RBVS MUST use the mapping between H.239 and BFCP messages shown in Table 4.1 to control the presentation token.

BFCP Message	H.239 Message	Purpose
FloorRequest	presentationTokenRequest	A request by the sender to acquire the indicated token.
FloorRequestStatus with an Accepted, Granted, Pending or Released status	presentationTokenResponse	Sent by the device receiving the presentationTokenRequest message either accepting or rejecting the request.
FloorRelease	presentationTokenRelease	Sent by a device holding the token in order to relinquish the token.
FloorQuery	presentationTokenIndicateOwner	Floor participants and floor chairs request information about a floor or floors by sending a FloorQuery message to the floor control server.
FloorStatus	presentationTokenIndicateOwner	Sent by the floor control server when the floor status changes or to parties that have requested status notification by sending a BFCP FloorQuery message to the floor control server.

Table 4.1 Mapping Between BFCP and H.239 Messages

4.2 Opening and Managing a Channel for Token Control Messages

4.2.1 Approach Used in H.239 for Transporting Token Control Messages

This section contains informative information only. It is included to provide background information regarding how H.239 commands are transported in H.323 systems.

H.245 is used to transport H.239 commands from one endpoint to another in H.323 systems. H.239 messages are GenericRequests, and H.239 responses are GenericResponses. In a point-to-point call, there is no need for one endpoint to act as a "master" for H.239 token control. An endpoint simply requests the token when it wishes to control presentation. However, in a centralized multipoint conference, it is necessary that the MCU control the token as the "master". In H.323, there is a concept of cascaded conferences, in which there are multiple MCUs and one MCU is designated as the "master". H.245 defines messages which enable the multiple MCUs to determine which one is the master. In a multipoint conference using H.239, the MCU is always the "master" which controls the token. In a cascaded MCU situation, the master MCU is always assumed to be the token control master. Thus in H.239 the MCU does all the token management. It accepts all token requests and grants/withdraws the token based on the situation. The MCU switches the video routing based on the token holder. When the endpoint receives an acknowledgement to the token request it will start sending content. The endpoint will also periodically send a ***presentationTokenIndicateOwner*** message which the MCU rebroadcasts to all the other endpoints. This ensures that all the endpoints are aware that content is being sent and from whom.

4.2.2 Best Practices in SIP for Transporting Token Control Messages

This section describes the methods SIP-based video endpoints are required to utilize to conform to RBVS "Best Practices" Profile. SIP-based video endpoints MUST use a UDP-based BFCP channel. The use of UDP to transport BFCP is preferred over TCP because existing NAT/firewall traversal solutions such as session border controllers (SBCs) or ICE can be utilized to enable the channel to traverse firewall. SIP-based video endpoints conforming to the "Best Practices" Profile for RBVS MUST utilize the following RFCs/drafts to open, manage and use a UDP-based BFCP channel to transport token control messages.

Function	Required RFC
Reliable transport of BFCP messages over UDP	draft-sandbakken-dispatch-bfcp-udp: Revision of the Binary Floor Control Protocol (BFCP) for use over an unreliable transport
Association of the control channel with one, or more, media channels	RFC 4583: Session Description Protocol (SDP) Format for Binary Floor Control Protocol (BFCP) Streams
Designation of which endpoint will function as the floor control server (token control master) and which will function as a BFCP client (token control slave)	RFC 4583: Session Description Protocol (SDP) Format for Binary Floor Control Protocol (BFCP) Streams
Conference ID and User IDs	RFC 4583: Session Description Protocol (SDP) Format for Binary Floor Control Protocol (BFCP) Streams

Table 4.2: Opening/Managing the BFCP Channel

Information used to open the BFCP channel is sent in the SDP offer and answer. SDP attributes are used to control the following aspects of the UDP–based BFCP channel:

1. The transport which will be used (i.e. UDP)
2. The media channel(s) which will be controlled by the BFCP channel
3. Which endpoint will function as the floor control server (token control master) and which will function as a BFCP client (token control slave)
4. The Conference ID and the User IDs

RFC 4583, "Session Description Protocol (SDP) Format for Binary Floor Control Protocol (BFCP) Streams", specifies how to accomplish these items for BFCP channels in general. IETF draft-sandbakken-dispatch-bfcp-udp , "Revision of the Binary Floor Control Protocol (BFCP) for use over an unreliable transport", extends this to cover UDP-based BFCP channels.
RFC 4583 defines attributes to describe BFCP streams in SDP session descriptions, enabling the connection to be negotiated via an offer/answer exchange. User agents typically use the offer/answer model to establish a number of media streams of different types. Following this model, a BFCP connection is described as an application media type stream by using an SDP 'm' line, possibly followed by a number of attributes encoded in 'a' lines.

Control Channel and Transport Type: An example of the 'm' line for a UDP-based BCFP channel is:

```
m = application 50000 UDP/BFCP *
```

Association of the Control Channel with Media Channels: Association between streams and floors is made using the floorid and label SDP media-level attributes as shown in the following example:

```
m=application 20000 UDP/BFCP *
a=confid:4321
a=userid:1234
a=floorid:1 mstrm:10
a=floorid:2 mstrm:11
a=floorctrl: c-only s-only
m=audio 21000 RTP/AVP 0
a=label:10
m=video 30000 RTP/AVP 31
a=label:11
```

Note: The ABNF for floorid is defined in RFC 4583 as follows:

```
floor-id-attribute = "a=floorid:" token [" mstrm:" token *(SP token)]
```

The text examples in RFC 4583 incorrectly use "m-stream" instead of "mstrm". RBVS implementations MUST follow the ABNF (i.e. use "mstrm") when encoding BFCP messages.

Floor Control Server or Client: RFC 4583 specifies the floorctrl attribute which enables an endpoint to declare that it is willing to act as the floor control server or a floor control client. This attribute is especially necessary for the multipoint case, when the MCU must act as the floor control server. It also enables endpoints to re-negotiate roles, such as in the case where a point-to-point call is converted to a multipoint call. For example, suppose EP_A and EP_B are in a call and that EP_B is the current floor server. Also suppose EP_A has MCU capabilities and a third endpoint, EP_C, calls into EP_A and joins the call. It is logical in this case to have the floor control server on EP_A. The floorctrl attribute enables the server role to be moved from EP_B to EP_A. In this scenario, EP_A will send a new offer to EP_B with the floor-control attribute set to "server-only".

The floorctrl attribute enables an endpoint to state, for a given BFCP stream, whether:

- c-only: The offerer is willing to act as a floor control client only
- s-only: The offerer is willing to act as a floor control server only
- c-s: The offerer would be willing to act both as a floor control client and as a floor control server

Note that a combination of attribute values may be used in an offer to indicate a willingness to take on any of a set of roles. In the answer, a single role is used to indicate the exact role of the offerer and the answerer. For example, an offer may indicate a willingness to be either a client or a server, e.g.:

a=floorctrl: c-only s-only

The answer indicates selected role; e.g.:

a=floorctrl: s-only

The result is the offerer is the BFCP client, and the answerer is the BFCP server.

The recommended usage of the floorctrl attribute for RBVS is as follows:

Offerers:

- ***Offerer is only a Client***: Offerer MUST offer "c-only".
- ***Offerer is only a Server***: Offerer MUST offer "s-only".
- ***Offerer can be either a Client or a Server***: Offerer MUST offer "c-only s-only" (or s-only c-only) since this conforms to RFC 4583. However, some implementations that can act as either a client or a server currently offer "c-s", but this is not recommended for RVBS.

Note that an offerer that can be a client and a server simultaneously can offer "c-only s-only c-s" (in any order). This is outside the scope of the use cases considered in this document, so it is not recommended behavior for RBVS.

Answerers:

- ***Offer is "c-only"***: If the answerer wants to act only as a server, it MUST answer with "s-only".
- ***Offer is "s-only"***: If the answerer wants to act only as a client, it MUST answer with "c-only".

- ***Offer is "c-only s-only"***: If the answerer wants to act only as a server, it MUST answer with "s-only". If the answerer wants to act only as a client, it MUST answer with "c-only".
- ***Offer is "c-s"***: The answerer SHOULD consider this the same offer as "c-only s-only" and SHOULD answer with "c-only" or "s-only". This is a deviation from RFC 4583 but is recommended as some implementations are known to offer "c-s" meaning "c-only s-only"
- ***Offer is "c-only s-only c-s"***: The answerer SHOULD answer with "c-only" or "s-only". The answerer can answer with "c-s" if it prefers to act as both a client and a server simultaneously; however, this is outside the scope of the use cases considered in this document so it is not recommended behavior for RVBS.

The possible offer/answer exchanges for the floorctrl attribute, the expected behavior for each exchange, and an indication of whether or not a particular answer is one recommended for RBVS in response to a particular offer, are given in Table 4.3:

A Offer to B	B Answer to A	Answer Recommended	Behavior	Remarks
c-only	c-only	No	BFCP Rejected	Both sides cannot be client
c-only	s-only	Yes	A is client, B is server	Expected if A can be client only
c-only	c-s	No	A is client, B is server	Recommended to answer s-only instead
c-only	c-only s-only	No	A is client, B is server	Recommended to answer s-only instead
c-only	c-only s-only c-s	No	A is client, B is server	Recommended to answer s-only instead
s-only	c-only	Yes	A is server, B is client	Expected if A can be server only
s-only	s-only	No	BFCP Rejected	Both sides cannot be server
s-only	c-s	No	A is server, B is client	Recommended to answer c-only instead
s-only	c-only s-only	No	A is server, B is client	Recommended to answer c-only instead
s-only	c-only s-only c-s	No	A is server, B is client	Recommended to answer c-only instead
c-s	c-only	Yes	A is server, B is client	Deviation from RFC 4583, but recommended for B to interwork with known implementations of A
c-s	s-only	Yes	A is client, B is server	Deviation from RFC 4583, but recommended for B to interwork with known implementations of A
c-s	c-s	No	Outside scope of RVBS	
c-s	c-only s-only	No	BFCP rejected	Client & server roles unclear
c-s	c-only s-only c-s	No	Outside scope of RVBS	
c-only s-only	c-only	Yes	A is server, B is client	
c-only s-only	s-only	Yes	A is client, B is server	
c-only s-only	c-s	No	BFCP rejected	Client & server roles unclear
c-only s-only	c-only s-only	No	BFCP rejected	Client & server roles unclear
c-only s-only	c-only s-only c-s	No	BFCP rejected	Client & server roles

				unclear
c-only s-only c-s	c-only	Yes	A is server, B is client	
c-only s-only c-s	s-only	Yes	A is client, B is server	
c-only s-only c-s	c-s	No	Outside scope of RVBS	
c-only s-only c-s	c-only s-only	No	BFCP rejected	Client & server roles unclear
c-only s-only c-s	c-only s-only c-s	No	BFCP rejected	Client & server roles unclear

Table 4.3: BFCP Channel floorctrl Attribute Exchange

Conference and User IDs: RFC 4583 also defines the confid and userid SDP media-level attributes which carry the integer representation of a conference ID and a user ID respectively:

- ***Conference ID***: this 32-bit field identifies the conference the BFCP message belongs to.
- ***User ID***: this field contains a 16-bit value that uniquely identifies a participant within a conference.

Endpoints acting as a BFCP client do not send conference ID and user ID attributes. Only the server includes these attributes in its SDP message body. Normally, in the initial offer, an endpoint will send the floorid attribute and will not send confid and userid attributes. The server will send those attributes back in the answer. There currently is no meaning for these two attributes. It is possible that in the future, they may play the same role as the MCU# and Terminal# in H.243.

Additional information on BFCP, specifically in regard to TCP-based BFCP channels, is provided in Appendix C.

5. Negotiating RBVS Support and Functionality

5.1 Approach Used in H.239 for RBVS Capability Negotiation

This section contains informative information only. It is included to provide background information regarding how H.239 capabilities are negotiated in H.323 systems.

H.323 systems use the H.245 capabilities exchange to indicate that they support H.239 during call setup. H.245 Terminal Capability Exchange is a procedure for exchanging preferred codecs and settings between the two H.323 terminals. The **h239ControlCapability** message indicates that the device supports H.239. A separate **h239ExtendedVideoCapability** message expresses video capabilities used with roles. The H.239 capability signals permit a device to send capabilities that correspond to the following:

- One or more capabilities for traditional video channel, and
- One or more capabilities for second video channel, with one or more capabilities for main video channel while second video channel is open

The **h239ExtendedVideoCapability** message binds together a set of alternative video channel capabilities for a single channel with the channel's capability to operate in one or more roles.

A role capability is not signaled for the traditional, main video channels. A role is signaled for the second video channel. Systems which support H.239 signal their capabilities according to these rules:

a) The traditional video channel is signaled normally.
b) A second video channel is signaled in an **ExtendedVideoCapability** containing a **videoCapability** and a **videoCapabilityExtension** containing the **h239ExtendedVideoCapability** and the roleLabel parameter. These signals mean that the device supports any of the roles indicated in the roleLabel parameter, on a video channel conforming to any of the indicated video capabilities.
c) The main video channel is included in a set of **simultaneousCapabilities** together with the **ExtendedVideoCapability** for the second video channel. This indicates that the main video channel may be used simultaneously with the second video channel.
d) The **h239ControlCapability** indicates that the device supports H.239 and the flowControlReleaseRequest and flowControlReleaseResponse messages.

5.2 Best Practices in SIP for RBVS Capabilities Negotiation

This section describes the methods SIP-based video endpoints are required to utilize to conform to the RBVS "Best Practices" Profile for negotiating role-based video stream capabilities using SDP offer/answer mechanisms. These methods enable the same level of functionality as described in Section 10.2 of H.239. RBVS offer/answer exchange:

1. Must not cause either a calling legacy endpoint or an answering legacy endpoint to perform in an unexpected manner. A legacy endpoint is defined as one that does not understand RBVS might not adhere to the multiple stream functionality specified in RFC 3264.
2. Should enable endpoints to use their "optimal" codec at all times. This includes when the endpoint is sending or receiving one, or more, video streams.
3. Should not require multiple streams to be opened until they are needed.
4. Must not preclude the use of multiple media streams in the other manners allowed by RFC 3264.

Offer/Answer procedures in which there are multiple streams of the same media type need to work in the following situations:

1. Endpoints negotiate multiple video streams as per RFC 3264 but with no content attributes. ***This is not a RBVS mode*** because there will be no ability for the roles of the channels to be designated or for endpoints to control presentation of any streams. That is, there is no specific applications semantics in this mode.

2. Endpoints negotiate multiple video streams as per RFC 3264, with content attributes, but with no BFCP control channel. The assignment of roles to streams can be very useful to aid endpoints in determining how to display the streams. But, it is ***not a RBVS mode*** if one of the streams is given the "slides" role and there is no associated BFCP channel because there will be no ability for the endpoints to control presentation of the "slides" stream.

3. An offering endpoint sends an offer for a single video stream with a content attribute along with a UDP-based BFCP channel containing a label for a stream which has not yet been offered. In this situation, an answerer which understands the "Best Practices" Profile of RBVS will accept the UDP-based BFCP channel and enter RBVS video operation. ***This is the RBVS "Best Practices" Profile mode*** because multiple streams will be ultimately offered, the roles of streams can be designated and the presentation of the stream associated with the BFCP channel can be controlled. In this scenario, the offer and acceptance of the BFCP channel is used as an indicator that both endpoints understand RBVS. The "presentation" video stream is not negotiated until it is needed. An endpoint that wishes to start presenting offers a "presentation" video stream, if one does not exist already. The endpoint may be an actual endpoint that is presenting or an MCU that is sending a presentation video stream on behalf of a participating endpoint. If the UDP-based BFCP channel is not accepted by the answerer, then the offerer MAY optionally offer a TCP-based BFCP channel if it wishes to try and interoperate with an older implementation which doesn't conform to the RVBS "Best Practices" Profile.

4. An offering endpoint sends an offer for multiple video streams, each with a content attribute, along with a UDP-based BFCP channel containing a label for one or more streams. In this situation, an answerer which understands the "Best Practices" Profile of RBVS may accept the UDP-based BFCP channel and enter RBVS video operation. ***This is not the RBVS "Best Practices" Profile mode*** because it may have backward compatibility issues with endpoints that do not understand RBVS. The "presentation" video stream is negotiated before it is needed.

When an endpoint that is capable of supporting SIP-based RBVS initiates a session in this mode, it MUST first offer an SDP which contains only one video stream and the UDP-based BFCP channel. It does this for three reasons:

1. ***To provide interoperability with legacy endpoints:*** the offering endpoint does not know if the second endpoint supports multiple streams, thus if the offering endpoint offers two streams, it has no guarantee that an answerer that supports only one stream, in selecting one of those streams, will select the stream that the offerer designated as the main stream. It is unlikely legacy endpoints will understand the content attribute. Endpoints that do not understand the content attribute might not automatically select the "main" stream as the single video stream.
2. ***To provide alternate capabilities for single/multiple video streams:*** it is desirable that endpoints can provide one set of capabilities (e.g. codecs and media stream bandwidths) for sessions when both "main" and "presentation" streams are used and a second set of capabilities for a session when only one stream (e.g. "main" stream) is used. For example, a PC-based endpoint may not be able to encode two streams using the H.264 video codec, but may have the processing power to encode one stream with H.264. Thus it may want to specify that H.263 codecs be used when "main" and "content" streams are negotiated, but use the H.264 codec when only the "main" stream is negotiated.
3. ***To avoid opening the "content" stream before it is needed:*** in nearly every call, users initiate the call, and then add "content" later in the call. Therefore, there is no reason to initiate multiple video channels at the beginning of the call and possibly use a less efficient codec for the "main" stream.

5.2.1 RBVS Compliant Negotiation

An example of an RBVS compliant offer and an RBVS answer is shown Figure 5.1. In this flow, the offerer initiates the call with an offer that includes one video channel and the UDP-based BFCP channel. An answerer that does not support RBVS will most likely send an answer which also includes a single video stream, but the answer will likely not include a BFCP channel. An answerer that does understand "Best Practices" RBVS, MUST send an answer containing one video channel (main) and the UDP-based BFCP channel.

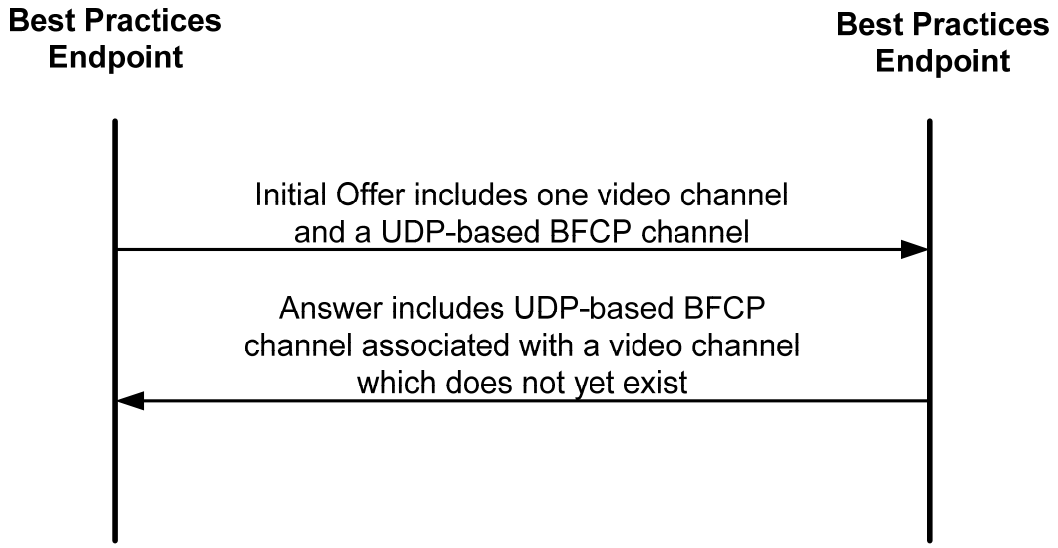

Figure 5.1 – RBVS Best Practice: "Best Practices" Endpoint Interoperating with another "Best Practices" Endpoint

After the initial session has been established, further negotiation will occur once there is a need for multiple streams (typically when a user wishes to display content). When this happens, one of the following sequences occurs depending on whether the requesting endpoint is the floor control client or floor control server:

- ***Floor Control Client:*** the requesting user's endpoint MUST send a BFCP floor request (i.e. requesting the token) to the floor control server, and the server MUST respond with a FloorRequestStatus message with a status of "granted", if the user is granted the token. The requesting user's endpoint MUST then initiate a re-INVITE with an offer containing a "main" stream and a "slides" stream. This is the case illustrated in Figure 5.2.
- ***Floor Control Server:*** server MUST send a FloorRequestStatus message with a status of "granted" to the other endpoint (which is the floor control client). The requesting endpoint MUST then initiate a re-INVITE with an offer containing a "main" stream and a "slides" stream.

A call flow illustrating complete RBVS compliant negotiation is provided in Figure 5.2.

Figure 5.2 – Call Flow for RBVS Negotiation Best Practice

In this profile, acceptance of the BFCP channel by the answerer is used to indicate that the answerer understands RBVS.

5.2.2 Interworking with Non-RBVS Implementations

If an RBVS compliant endpoint receives an offer containing a single video channel and TCP-based BFCP, it can optionally accept the offer. Similarly, if an RBVS compliant endpoint makes an offer and UDP-based BFCP is rejected, it can optionally attempt to offer TCP-based BFCP. In either case, if the TCP-based BFCP channel works, it may be possible to operate RBVS as if the BFCP channel were UDP-based, but this is outside the scope of RBVS "Best Practices".
Figure 5.3 shows how an endpoint operating within the “Best Practice” Profile will negotiate RBVS with: a) non-RBVS endpoints that do not support BFCP, and b) non-RBVS endpoints which support TCP-based BFCP instead of UDP-based BFCP.

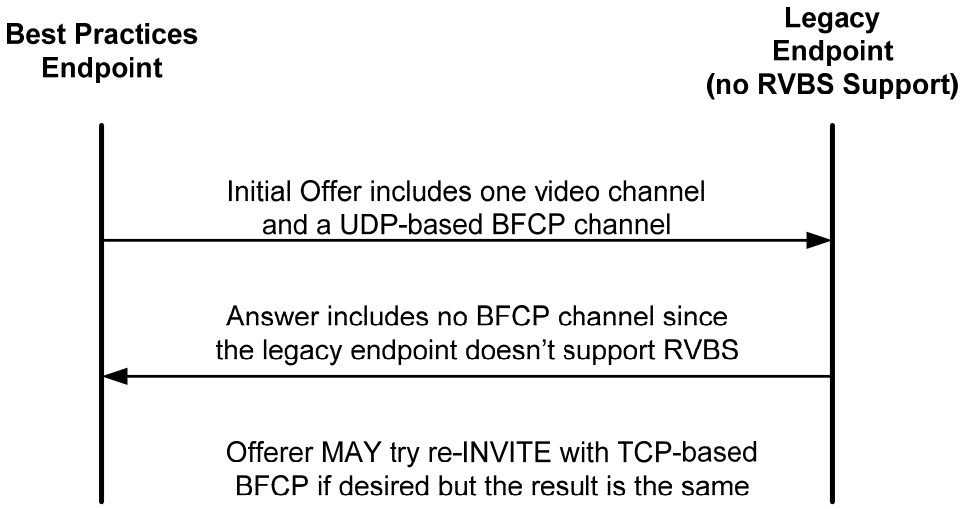

Figure 5.3a – RBVS Best Practice: “Best Practices” Endpoint Interoperating with non-RBVS Endpoint that does not Support BFCP

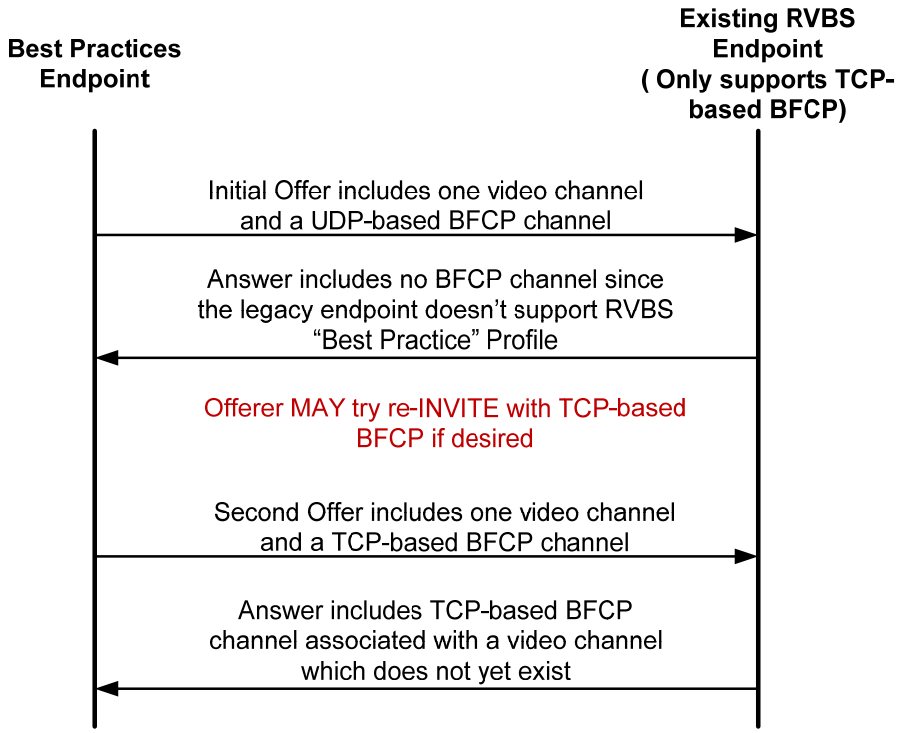

Figure 5.3b – RBVS Best Practice: "Best Practices" Endpoint Interoperating with non-RBVS Endpoint supporting TCP-based BFCP instead of UDP-based BFCP

5.2.3 Summary of Best Practice Offer/Answer Behavior

In table 5.1, each row shows a different initial SDP answer that might be received in response to an SDP offer that complies with RBVS "Best Practices", thereby containing a single video channel and a UDP-based BFCP channel. For each answer, the second column shows how this is to be interpreted, and the third column shows how the offerer should behave.

Answer received	Interpretation of answer	Required behavior of offerer
UDP-based BFCP and single video channel both accepted.	The answerer complies with RBVS "Best Practices".	Continue in accordance with RBVS "Best Practices".
UDP-based BFCP rejected and single video channel accepted.	The answerer does not comply with RBVS "Best Practices", but might support RBVS using TCP-based BFCP.	Optionally, repeat the offer with TCP-based BFCP instead of UDP-based BFCP. If accepted, and if the BFCP channel works, it may be possible to operate RBVS as if the BFCP channel were UDP-based, but this is outside the scope of RBVS "Best Practices". Other possible actions are outside the scope of RBVS "Best

		Practices", e.g., continue without BFCP or abandon the call.
UDP-based BFCP accepted and single video channel rejected.	The answerer might comply with RBVS "Best Practices" if an acceptable proposal for a video channel can be found	Possible actions are outside the scope of RBVS "Best Practices", e.g., repeat the offer with different video parameters, continue without video, or abandon the call.
UDP-based BFCP rejected and single video channel rejected	The answerer does not comply with RBVS "Best Practices"	Possible actions are outside the scope of RBVS "Best Practices"

Table 5.1 - RBVS "Best Practices" offerer behavior to different answers to an initial offer

In table 5.2, each row shows a different initial SDP offer that might be received by an answerer that complies with RBVS "Best Practices". For each offer, the second column shows how this is to be interpreted, and the third column shows how the answerer should behave.

Offer received	Interpretation of offer	Required behavior of answerer
UDP-based BFCP and single video channel.	The offerer complies with RBVS "Best Practices".	Continue in accordance with RBVS "Best Practices".
UDP-based BFCP and multiple video channels..	The offerer does not comply with RBVS "Best Practices".	Optionally, accept the offer and operate RBVS as if only a single video channel had been offered initially and the other video channel(s) had been added later; but this, and other possible actions, are outside the scope of RBVS "Best Practices".
UDP-based BFCP and no video channel.	The offerer does not comply with RBVS "Best Practices".	Possible actions are outside the scope of RBVS "Best Practices".
TCP-based BFCP and single video channel.	The offerer does not comply with RBVS "Best Practices" but might support RBVS using TCP-based BFCP.	Optionally, accept the offer, and if the TCP-based BFCP channel works, it may be possible to operate RBVS as if the BFCP channel were UDP-based, but this is outside the scope of RBVS "Best Practices". Other possible actions are outside the scope of RBVS "Best Practices", e.g., reject BFCP or abandon the call.
TCP-based BFCP and multiple video channels..	The offerer does not comply with RBVS "Best Practices".	Possible actions are outside the scope of RBVS "Best Practices".
TCP-based BFCP and no video channel.	The offerer does not comply with RBVS "Best Practices".	Possible actions are outside the scope of RBVS "Best Practices".

No BFCP and single video channel.	The offerer does not comply with RBVS "Best Practices".	Possible actions are outside the scope of RBVS "Best Practices".
No BFCP and multiple video channels..	The offerer does not comply with RBVS "Best Practices".	Possible actions are outside the scope of RBVS "Best Practices".

Table 5.2 - RBVS "Best Practices" answerer behavior to different initial offers

6. Firewall Traversal Issues

6.1 BFCP Token Control Channel

Implementations should follow the recommendations provided in draft-sandbakken-dispatch-bfcp-udp.

6.2 Media Channels

Because endpoints send content only when they have the presentation token, intermediate NATs in the media path may close bindings which would close down the presentation (content) media channel.

The ICE specification (RFC 5245) already addresses the problem of keeping NAT bindings alive, primarily when dealing with media channels which are put “on-hold”. The same mechanisms SHOULD be used for the presentation channel in RBVS. In general, for UDP media streams, if both sides support ICE, then the STUN Binding request is used to keep the channel alive. If one of the sides does not support ICE, an endpoint SHOULD use one of the following mechanisms:

1. Send RTP packets with a payload type supported by the channel but with a zero-length payload approximately once every 30 seconds when the endpoint does not hold the presentation token.
2. Send RTP no-ops on media channels approximately once every 30 seconds when the endpoint does not hold the presentation token. No-ops should not be sent unless the receiving device has declared support for receiving no-op RTP packets.
3. Use an RTP keep-alive method described in RFC 6263. The recommended method requires support for RTP/RTCP multiplexing by both sides. RTP/RTCP multiplexing is outside the scope of this best practice; therefore, in the event the other side does not support it, other methods identified in RFC 6263 may be used.

7. Relevant RFCs and Drafts

RFC	Title	Usage
RFC 4145	TCP-Based Media Transport in the Session Description Protocol (SDP)	Describes how to express media transport over TCP using the Session Description Protocol (SDP)
RFC 4572	Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP)	Specifies how to establish secure connection-oriented media transport sessions over the Transport Layer Security (TLS) protocol using the

		Session Description Protocol (SDP)
RFC 4574	The Session Description protocol (SDP) Label Attribute	Defines a new Session Description Protocol (SDP) media-level attribute: "label".
RFC 4582	The Binary Floor Control Protocol (BFCP)	Specifies the Binary Floor Control Protocol (BFCP) which is used for token control
RFC 4583	Session Description Protocol (SDP) Format for Binary Floor Control Protocol (BFCP) Streams	Specifies how to describe BFCP streams in Session Description Protocol (SDP) descriptions.
RFC 4796	The SDP (Session Description Protocol) Content Attribute	Defines a new Session Description Protocol (SDP) media-level attribute, 'content'.
RFC 5018	Connection Establishment in the Binary Floor Control Protocol (BFCP)	Specifies how a Binary Floor Control Protocol BFCP client establishes a connection to a BFCP floor control server outside the context of an offer/answer exchange.
RFC 5245	Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols	Describes a protocol for Network Address Translator (NAT) traversal for UDP-based multimedia sessions established with the offer/answer model.
RFC 6263	Application Mechanism for maintaining alive the Network Address Translator (NAT) mappings associated to RTP flows	Lists the different mechanisms that enable applications using Real-time Transport Protocol (RTP) to maintain their RTP Network Address Translator (NAT) mappings alive. It also makes a recommendation for a preferred mechanism.
draft-sandbakken-dispatch-bfcp-udp	Revision of the Binary Floor Control Protocol (BFCP) for use over an unreliable transport	Extends the Binary Floor Control Protocol enabling it to use UDP as a transport.

Appendix A: BFCP Background

The Binary Floor Control Protocol (BFCP) is defined in RFC 4582. Floor control is a means to manage joint or exclusive access to shared resources in a (multiparty) conferencing environment. In Role Based Video Streams, BFCP is used to control the role of presenter for the presentation channel.

The Binary Floor Control Protocol (BFCP) utilizes the following terminology:

- ***Floor:*** A permission to temporarily access or manipulate a specific shared resource or set of resources.
- ***Floor chair:*** A user (or an entity) who manages one floor (grants, denies, or revokes a floor). The floor chair does not have to be a member in a conference.
- ***Floor control:*** A mechanism that enables applications or users to gain safe and mutually exclusive or non-exclusive input access to the shared object or resource.
- ***Floor control server:*** A logical entity that maintains the state of the floor(s) including which floors exists, who the floor chairs are, who holds a floor, etc. Requests to manipulate a floor are directed at the floor control server.

Floors are associated with resources. For example, a floor that controls who talks at a given time, has a particular audio stream as its associated resource. Associations between floors and streams are made using the mechanisms described in Session Description Protocol (SDP) Format for Binary Floor Control Protocol (BFCP) Streams (RFC 4583).

A floor participant creates a connection to the floor control server based on the SDP negotiation described in Section 5.2 and Appendix C. A party that wants the floor will send a floor request to the floor control server. The floor request includes the conference ID, user ID and floor ID from the SDP. The floor control server accepts the requests and sends back a FloorRequestStatus message as a response, the status may be granted if the floor is available or pending if the server needs revoke it from a different user. A floor request is considered on-going as long as it is not in the cancelled, released or revoked state. The server sends FloorRequestStatus to all participants when the state changes. It may continue to send FloorRequestStatus periodically.

If the floor is held by another participant, the floor control server may send that participant a FloorRequestStatus with state revoke. The participant holding the floor will send a FloorRelease message to the Floor control server. The floor control server acknowledges using FloorRequestStatus with Released status and sends the participant who requested the floor a FloorRequestStatus with granted status. The FloorRequestStatus may be sent to all parties who are waiting for status on either FloorRequests or parties who sent FloorQuery messages.

Appendix B: Overview of TCP-Based BFCP RBVS Implementations

As of September 2010, the most widely deployed method for creating a channel to transport BFCP messages for controlling the presenter is to use a TCP channel between a floor control client and a floor control server. Since this TCP channel is described in the SDP, it essentially appears to be a TCP-based media stream as far as entities such as application-layer gateways and session border controllers are concerned. TCP-based BFCP cannot be successfully used with many existing firewall/NAT traversal solutions. That is why the "Best Practices" Profile described in this document specifies the use of an UDP-based BFCP channel.

In most TCP-based BFCP implementations, SIP-based video endpoints use the RFCs listed in Table B.1 to open, manage and use a TCP-based BFCP channel to transport token control messages. Note that TLS could be used to secure the BFCP-over TCP channel, but there are no known implementations that secure TCP-based BFCP channels.

Function	Required RFC
Indication of which transport will be used (TCP or TLS-over-TCP)	RFC 4583: Session Description Protocol (SDP) Format for Binary Floor Control Protocol (BFCP) Streams
Association of the control channel with one, or more, media channels	RFC 4583: Session Description Protocol (SDP) Format for Binary Floor Control Protocol (BFCP) Streams
Designation of which endpoint will function as the floor control server (token control master) and which will function as a BFCP client (token control slave)	RFC 4583: Session Description Protocol (SDP) Format for Binary Floor Control Protocol (BFCP) Streams
Management of the TCP/TLS connection, including: • Which endpoint should initiate the connection, • The port the endpoints listen on, • Whether the connection is new or an existing connection	RFC 4145: TCP-Based Media Transport in the Session Description Protocol (SDP) RFC 4572: Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP)
Conference ID and User IDs	RFC 4583: Session Description Protocol (SDP) Format for Binary Floor Control Protocol (BFCP) Streams

Table B.1: Opening/Managing the TCP-based BFCP Channel

Appendix C provides an informational background for RFC 4583 and relates it to RBVS. It also provides informational background for RFC 4145. A typical call flow in a TCP-based BFCP implementation is shown in Figure B.1.

Figure B.1 – Typical Call Flow in TCP-based BFCP RBVS Implementations

Appendix C: Controlling and Managing a TCP-Based BFCP Channel

Information used to open the BFCP channel is sent in the SDP offer and answer. SDP attributes are used to control the following aspects of the BFCP channel:

1. The transport which will be used (TCP, TLS-over-TCP or UDP),
2. The media channel(s) which will be controlled by the BFCP channel,
3. Designate which endpoint will function as the floor control server (token control master) and which will function as a BFCP client (token control slave),
4. The Conference ID and the User IDs
5. Management of the connection. This information includes:
    a. Which endpoint should initiate the connection (only when TCP is used as the transport),
    b. The port the endpoints listen on,
    c. Whether the connection is new or an existing connection

RFC 4583 ("Session Description Protocol (SDP) Format for Binary Floor Control Protocol (BFCP) Streams") specifies how to accomplish the first four of these items. Specific recommendations for these are provided in section 4 of this specification. The management of the connection is described in RFC 4145

Functionality specified in RFC 4583 is used to describe BFCP streams in SDP session descriptions to enable the connection to be established via an offer/answer exchange. User agents typically use the offer/answer model to establish a number of media streams of different types. Following this model, a BFCP connection is described as an application media type stream by using an SDP 'm' line, possibly followed by a number of attributes encoded in 'a' lines.

Transport Type: RFC 4583 defines two values for the transport field for the "m" line: TCP/BFCP and TCP/TLS/BFCP. The former is used when BFCP runs directly on top of TCP and the latter is used when BFCP runs on top of TLS, which in turn runs on top of TCP. The fmt (format) list is ignored for BFCP. The fmt list of BFCP m lines should contain a single "*" character. The following is an example of an m line for a BFCP connection:

```
m=application 20000 TCP/BFCP *
```

Association of the Control Channel with Media Channels: Association between streams and floors is made using the floorid and label SDP media-level attributes as shown here:

```
m=application 20000 TCP/BFCP *
a=setup:actpass
a=connection:new
a=confid:4321
a=userid:1234
a=floorid:1 mstrm:10
a=floorid:2 mstrm:11
a=floorctrl: c-only s-only
m=audio 21000 RTP/AVP 0
a=label:10
```

m=video 30000 RTP/AVP 31
a=label:11

Floor Control Server or Client: RFC 4583 also specifies the floorctrl attribute which enables an endpoint to declare that it needs to act as the floor control server or a floor control client. This attribute is especially necessary for the multipoint case, when the MCU must act as the floor control server. It also enables endpoints to re-negotiate roles, such as in the case where a point-to-point call is converted to a multipoint call. For example, suppose EP_A and EP_B are in a call and that EP_B is the current floor server. Also suppose EP_A has MCU capabilities and a third endpoint, EP_C, calls into EP_A and joins the call. It is logical in this case to have the floor control server on EP_A. The floorctrl attribute enables the server role to be moved from EP_B to EP_A. In this scenario, EP_A will send a new offer to EP_B with the floor-control attribute set to "server-only".

The recommended usage of the floorctrl attribute is covered in section 4.

Connection Management: When using connection based transport, the management of the connection used to transport BFCP is performed using the 'setup', port and 'connection' attributes as defined in RFC 4145.

Setup Attribute: The setup attribute indicates which of the end points should initiate the connection establishment (e.g., send the initial TCP SYN). The setup attribute has these values:

- *active*: The endpoint will initiate an outgoing connection.
- *passive*: The endpoint will accept an incoming connection.
- *actpass*: The endpoint is willing to accept an incoming connection or to initiate an outgoing connection.
- *holdconn*: The endpoint does not want the connection to be established for the time being.

For BFCP channels, all entities SHOULD use actpass so that the likelihood of achieving firewall traversal is higher when endpoints such as session border controllers are in place. However, if an endpoint knows, for sure, that it has a globally-routable IP address, then it can use passive for the setup attribute.

Port Number: For BFCP channels, all entities indicate actpass and need to designate the port at which they will be listening for a connection. Each endpoint MUST initiate a connection to the port number on the m= line in the SDP from the other endpoint. In addition, endpoints MUST initiate the connection from the same port they designated in the m= line of the SDP that they sent. This symmetry will enable a mapping to be set up in firewall/NATs and allow traversal to occur when session border controllers are utilized.

In the case of TCP-based BFCP, since both endpoints use actpass, both will attempt to open a TCP connection (this is the so-called simultaneous open in TCP). While TCP stacks are designed to handle this and a single TCP connection will result, NAT treatment of simultaneous opens is currently not well defined, though specifications are being developed to address this. Some NATs generate a reset upon receipt of the second TCP SYN packet, which will cause the connection attempt to fail. In that case, the TCP connection attempt will timeout.

The Connection Attribute: SDP may be exchanged between endpoints at various stages of a session to accomplish tasks such as terminating a session, redirecting media to a new endpoint, or renegotiating the media parameters for a session. After the initial session has been established, it may be ambiguous as to whether subsequent SDP exchange represents a confirmation that the endpoint is to continue using the current BFCP connection unchanged, or is a request to make a new media connection. The media-level connection attribute is used to disambiguate these two scenarios.

Offerers and answerers use the connection attribute to decide whether a new transport connection needs to be established or, on the other hand, the existing transport connection should still be used. The connection value resulting from an offer/answer exchange is the connection value in the answer. If the connection value in the answer is "new", the end-points should establish a new connection. If the connection value in the answer is "existing", the end-points should continue using the existing connection.

Examples: The following is an example of an offer sent by a conference floor server to a BFCP client for a TCP-based BFCP connection. For the purpose of brevity, the main portion of the session description is omitted in the examples, which only show m= lines and their attributes.

```
m=application 20000 TCP/BFCP *
a=setup:actpass
a=connection:new
a=confid:4321
a=userid:1234
a=floorid:1 mstrm:10
a=floorid:2 mstrm:11
a=floorctrl: c-only s-only
m=audio 21000 RTP/AVP 0
a=label:10
m=video 30000 RTP/AVP 31
a=label:11
```

The following is the answer returned by the client.

```
m=application 21000 TCP/BFCP *
a=setup:actpass
a=connection:new
a=confid:4321
a=userid:4444
a=floorctrl: c-only
m=audio 25000 RTP/AVP 0
m=video 35000 RTP/AVP 31
```

Note that floor control messages for multiple streams (i.e. multiple floorids) can be transported on the same BFCP channel.