BFCPbis Working Group
Internet-Draft
Obsoletes: 4583 (if approved)
Intended status: Standards Track
Expires: September 23, 2018

C. Holmberg
G. Camarillo
Ericsson
T. Kristensen
Cisco

March 22, 2018

Session Description Protocol (SDP) Format for Binary Floor Control Protocol (BFCP) Streams
draft-ietf-bfcpbis-rfc4583bis-19

Abstract

This document defines the Session Description Protocol (SDP) offer/answer procedures for negotiating and establishing Binary Floor Control Protocol (BFCP) streams.

This document obsoletes RFC 4583. Changes from RFC 4583 are summarized in Section 15.

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 https://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 September 23, 2018.

Copyright Notice

Copyright (c) 2018 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 (https://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

As discussed in the BFCP (Binary Floor Control Protocol) specification [7], a given BFCP client needs a set of data in order to establish a BFCP connection to a floor control server. This data includes the transport address of the server, the conference identifier, and the user identifier.

One way for clients to obtain this information is to use an SDP offer/answer [4] exchange. This document specifies how to encode this information in the SDP session descriptions that are part of such 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 any other media stream by using an SDP 'm' line, possibly followed by a number of attributes encoded in 'a' lines.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [1] and indicate requirement levels for compliant implementations.

3. Fields in the 'm' Line

This section describes how to generate an 'm' line for a BFCP stream.

According to the SDP specification [10], the 'm' line format is the following:

```
m=<media> <port> <proto> <fmt> ...
```

The media field MUST have a value of "application".

The port field is set depending on the value of the proto field, as explained below. A port field value of zero has the standard SDP meaning (i.e., rejection of the media stream) regardless of the proto field.

When TCP is used as the transport, the port field is set following the rules in [6]. Depending on the value of the 'setup' attribute (discussed in Section 10.1), the port field contains the port to which the remote endpoint will direct BFCP messages, or in the case where the endpoint will initiate the connection towards the remote endpoint, should be set to a value of 9.

When UDP is used as the transport, the port field contains the port to which the remote endpoint will direct BFCP messages regardless of the value of the 'setup' attribute.

This document defines five values for the proto field: TCP/BFCP, TCP/DTLS/BFCP, TCP/TLS/BFCP, UDP/BFCP, and UDP/TLS/BFCP.

TCP/BFCP is used when BFCP runs directly on top of TCP. TCP/TLS/BFCP is used when BFCP runs on top of TLS, which in turn runs on top of TCP. TCP/DTLS/BFCP is used when running BFCP on top of DTLS [11], as described in this specification, which in turn runs on top of TCP using the framing method defined in [12] with DTLS packets being sent and received instead of RTP/RTCP packets using the shim defined in RFC4571 such that the length field defined in RFC4571 precedes each DTLS message.

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, which in turn runs on top of UDP.

The fmt (format) list is not applicable to BFCP. The fmt list of 'm' lines in the case of any proto field value related to BFCP MUST contain a single "*" character. If the the fmt list contains any other value it is ignored.

The following is an example of an 'm' line for a BFCP connection:

```
m=application 50000 TCP/TLS/BFCP *
```

4. Floor Control Roles

When two endpoints establish a BFCP stream, they need to determine which of them acts as floor control client and which acts as floor control server. Typically, a client that establishes a BFCP stream with a conference server will act as floor control client, while the conference server will act as floor control server. However, there are scenarios where both endpoints would be able to act as floor control server. For example, in a two-party session that involves an audio stream and a shared whiteboard, the endpoints need to determine which party will be act as floor control server.

Furthermore, there are situations where both endpoints act as both floor control client and floor control server within the same session. For example, in a two-party session that involves an audio stream and a shared whiteboard, one endpoint acts as the floor control server for the audio stream and the other endpoint acts as the floor control server for the shared whiteboard. However, for a given media stream one endpoint MUST act as floor control client and one endpoint MUST act as floor control server.

5. SDP 'floorctrl' Attribute

This section defines the SDP 'floorctrl' media-level attribute. The attribute is used to determine the floor control role for a given stream (represented by an SDP media description).

The Augmented BNF syntax [2] for the attribute is:

```
floor-control-attribute  = "a=floorctrl:" role *(SP role)
role                     = "c-only" / "s-only" / "c-s"
```

An endpoint includes the attribute to indicate the role(s) it would be willing to perform for the stream associated with the attribute:

c-only: The endpoint is willing to act as floor control client.

s-only: The endpoint is willing to act as floor control server only.

c-s: The endpoint is willing to act as floor control client and floor control server.

An offerer can indicate multiple attribute value for a given session. An answerer can only indicate one attribute value. The offerer indicates which floor control role(s) that it is willing to take. The answerer indicates the role taken by the answerer. Based on this, the floor control of the offerer is determined, as shown in Table 1.

Offerer	Answerer
c-only	s-only
s-only	c-only
c-s	c-s

Table 1: Roles

Endpoints compliant with [16] might not include the 'floorctrl' attribute in offers and answerer. For a given stream, if the 'floorctrl' attribute is not present the offerer will act as floor control client and the answerer will act as floor control server.

The SDP Offer/Answer procedures for the 'floorctrl' attribute are defined in Section 13.

The following is an example of a 'floorctrl' attribute in an offer:

```
a=floorctrl:c-only s-only c-s
```

6. SDP 'confid' and 'userid' Attributes

This section defines the SDP 'confid' and the 'userid' media-level attributes. The attributes are used by a floor control server to convey the conference ID value and user ID value to the floor control client, using decimal integer representation.

The Augmented BNF syntax [2] for the attributes is:

```
confid-attribute      = "a=confid:" conference-id
conference-id         = token
userid-attribute      = "a=userid:" user-id
user-id               = token

token-char            = %x21 / %x23-27 / %x2A-2B / %x2D-2E / %x30-39
                        / %x41-5A / %x5E-7E
token                 = 1*(token-char)
```

;token-char and token elements are defined in [RFC4566].

The SDP Offer/Answer procedures for the 'confid' and 'userid' attributes are defined in Section 13.

7. SDP 'floorid' Attribute

This section defines the SDP 'floorid' media-level attribute. The attribute conveys a floor identifier, and optionally pointers to one or more BFCP-controlled media streams.

The Augmented BNF syntax [2] for the attribute is:

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

The floor identifier value is the integer representation of the Floor ID to be used in BFCP. Each media stream pointer value is associated with an SDP 'label' attribute [8] of a media stream.

The SDP Offer/Answer procedures for the 'floorid' attribute are defined in Section 13.

Note: In [16] 'm-stream' was erroneously used in Section 14. Although the example was non-normative, it is implemented by some vendors and occurs in cases where the endpoint is willing to act as an server. Therefore, it is RECOMMENDED to support parsing and interpreting 'm-stream' the same way as 'mstrm' when receiving.

8. SDP 'bfcpver' Attribute

This section defines the SDP 'bfcpver' media-level attribute. The attribute is used to negotiate the BFCP version.

The Augmented BNF syntax [2] for the attributes is:

```
bfcp-version-attribute = "a=bfcpver:" bfcp-version *(SP bfcp-version)
bfcp-version           = token
```

An endpoint uses the 'bfcpver' attribute to convey the version(s) of BFCP supported by the endpoint, using integer values. For a given version, the attribute value representing the version MUST match the "Version" field that would be presented in the BFCP COMMON-HEADER [7]. The BFCP version that will eventually be used will be conveyed with a BFCP-level Hello/HelloAck.

Endpoints compliant with [16] might not always include the 'bfcpver' attribute in offers and answers. For a given stream, if the 'bfcpver' attribute is not present, the default values are inferred from the transport specified in the 'm' line (Section 3) associated with the stream. In accordance with definition of the Version field in [7], when used over a reliable transport the default attribute value is "1", and when used over an unreliable transport the default attribute value is "2".

The SDP Offer/Answer procedures for the 'bfcpver' attribute are defined in Section 13.

9. Multiplexing Considerations

[20] defines how multiplexing of multiple media streams can be negotiated. This specification does not define how BFCP streams can be multiplexed with other media streams. Therefore, a BFCP stream MUST NOT be associated with a BUNDLE group [20]. Note that BFCP-controlled media streams might be multiplexed with other media streams.

[21] defines the mux categories for the SDP attributes defined in this specification, excluding the SDP 'bfcpver' attribute. . Table 2 defines the mux category for the 'bfcpver' attribute:

Name	Notes	Level	Mux Category
bfcpver	Needs further analysis	M	TBD

Table 2: Multiplexing Attribute Analysis

10. BFCP Connection Management

BFCP streams can use TCP or UDP as the underlying transport. Endpoints exchanging BFCP messages over UDP send the BFCP messages towards the peer using the connection address and port provided in the SDP 'c' and 'm' lines. TCP connection management is more complicated and is described in the following Section.

> Note: When using Interactive Connectivity Establishment (ICE) [17], TCP/DTLS/BFCP, and UDP/TLS/BFCP, the straight-forward procedures for connection management as UDP/BFCP described above apply. TCP/TLS/BFCP follows the same procedures as TCP/BFCP and is described below.

10.1. TCP Connection Management

The management of the TCP connection used to transport BFCP messages is performed using the SDP 'setup' and 'connection' attributes [6]. The 'setup' attribute indicates which of the endpoints initiates the TCP connection. The 'connection' attribute handles TCP connection re-establishment.

The BFCP specification [7] describes a number of situations when the TCP connection between a floor control client and the floor control server needs to be re-established. However, that specification does not describe the re-establishment process because this process depends on how the connection was established in the first place. Endpoints using the offer/answer mechanism follow the following rules.

When the existing TCP connection is closed and re-established following the rules in [7], the floor control client MUST send an offer towards the floor control server in order to re-establish the connection. If a TCP connection cannot deliver a BFCP message and times out, the endpoint that attempted to send the message (i.e., the one that detected the TCP timeout) MUST send an offer in order to re-establish the TCP connection.

Endpoints that use the offer/answer mechanism to negotiate TCP connections MUST support the 'setup' and 'connection' attributes.

11. Authentication

When a BFCP stream is negotiated using the SDP offer/answer mechanism, it is assumed that the offerer and the answerer authenticate each other using some mechanism. TLS/DTLS is the preferred mechanism. Other mechanisms are possible, but are outside the scope of this document. Once this mutual authentication takes place, all the offerer and the answerer need to ensure is that the entity they are receiving BFCP messages from is the same as the one that generated the previous offer or answer.

The initial mutual authentication SHOULD take place at the signaling level. Additionally, signaling can use S/MIME [5] to provide an integrity-protected channel with optional confidentiality for the offer/answer exchange. BFCP takes advantage of this integrity-protected offer/answer exchange to perform authentication. Within the offer/answer exchange, the offerer and answerer exchange the fingerprints of their self-signed certificates. These self-signed certificates are then used to establish the TLS/DTLS connection that will carry BFCP traffic between the offerer and the answerer.

Endpoints follow the rules in [9] regarding certificate choice and presentation. Endpoints that use the offer/answer model to establish BFCP streams MUST support the 'fingerprint' attribute and MUST include it in their offers and answers.

When TLS is used with TCP, once the underlying connection is established, the answerer, which can be the floor control client or the floor control server, acts as the TLS server regardless of its role (passive or active) in the TCP establishment procedure. If the TCP connection is lost, the active endpoint is responsible for re-establishing the TCP connection. Unless a new TLS session is negotiated, subsequent SDP offers and answers will not impact the previously negotiated TLS roles.

When DTLS is used with UDP, the requirements specified in Section 5 of [14] MUST be followed.

> Informational note: How to determine which endpoint initiates the TLS/DTLS association depends on the selected underlying transport. It was decided to keep the original semantics in [16] for TCP to retain backwards compatibility. When using UDP, the procedure defined in [14] was selected in order to be compatible with other DTLS based protocol implementations, such as DTLS-SRTP. Furthermore, the procedure defined in [14] do not overload offer/answer semantics and works for offerless INVITE in scenarios with B2BUAs.

12. ICE Considerations

Generic SDP offer/answer procedures for Interactive Connectivity Establishment (ICE) are defined in [18].

When BFCP is used with UDP based ICE candidates [17] then the procedures for UDP/TLS/BFCP are used.

When BFCP is used with TCP based ICE candidates [13] then the procedures for TCP/DTLS/BFCP are used.

Based on the procedures defined in [14], endpoints treat all ICE candidate pairs associated with a BFCP stream on top of a DTLS association as part of the same DTLS association. Thus, there will only be one BFCP handshake and one DTLS handshake even if there are multiple valid candidate pairs, and if BFCF media is shifted between candidate pairs (including switching between UDP to TCP candidate pairs) prior to nomination. If new candidates are added, they will also be part of the same DTLS association.

In order to maximize the likelihood of interoperability between the endpoints, all ICE enabled BFCP-over-DTLS endpoints SHOULD implement support for UDP/TLS/BFCP.

When an SDP offer or answer conveys multiple ICE candidates for a BFCP stream, UDP based candidates SHOULD be included and the default candidate SHOULD be chosen from one of those UDP candidates. If UDP transport is used for the default candidate, then the 'm' line proto value MUST be 'UDP/TLS/BFCP'. If TCP transport is used for the default candidate, the 'm' line proto value MUST be 'TCP/DTLS/BFCP'.

> Note: Usage of ICE with protocols other than UDP/TLS/BFCP and TCP/DTLS/BFCP is outside of scope for this specification.

13. SDP Offer/Answer Procedures

This section defines the SDP offer/answer [4] procedures for negotiating and establishing a BFCP stream. Generic procedures for DTLS are defined in [14]. Generic procedures for TLS are defined in [9].

This section only defines the BFCP-specific procedures. The procedures apply to an 'm' line describing a BFCP stream. If an offer or answer contains multiple 'm' lines describing BFCP streams, the procedures are applied independently to each stream.

If the 'm' line 'proto' value is 'TCP/TLS/BFCP', 'TCP/DTLS/BFCP' or 'UDP/TLS/BFCP', the offerer and answerer follow the generic procedures defined in [9].

If the 'm' line proto value is 'TCP/BFCP', 'TCP/TLS/BFCP', 'TCP/DTLS/TCP' or 'UDP/TLS/BFCP', the offerer and answerer use the SDP 'setup' attribute according to the procedures in [6].

If the 'm' line proto value is 'TCP/BFCP', 'TCP/TLS/BFCP' or 'TCP/DTLS/BFCP', the offerer and anwerer use the SDP 'connection' attribute according to the procedures in [6].

Note: The use of source-specific SDP parameters [19] is not defined to BFCP streams.

13.1. Generating the Initial SDP Offer

When the offerer creates an initial offer, the offerer:

- MUST associate an SDP 'floorctrl' attribute (Section 5), with the 'm' line;
- MUST associate an SDP 'confid' attribute (Section 6), with the 'm' line;
- MUST associate an SDP 'userid' attribute (Section 6), with the 'm' line;
- MUST associate an SDP 'floorid' attribute (Section 7), with the 'm' line;
- MUST associate an SDP 'label' attribute (Section 7), with the 'm' line; and
- MUST associate an SDP 'bfcpver' attribute (Section 8), with the 'm' line.

13.2. Generating the SDP Answer

When the answerer receives an offer, which contains an 'm' line describing a BFCP stream, if the answerer accepts the 'm' line it:

- MUST insert a corresponding 'm' line in the answer, with an identical 'm' line proto value [4]
- MUST, if the offer contained an SDP 'floorctrl' attribute, associate a 'floorctrl' attribute with the 'm' line;

- MUST associate an SDP 'confid' attribute with the 'm' line;
- MUST associate an SDP 'userid' attribute with the 'm' line;
- MUST associate an SDP 'floorid' attribute with the 'm' line;
- MUST associate an SDP 'label' attribute with the 'm' line; and
- MUST, if the offer contained an SDP 'bfcpver' attribute, associate a 'bfcpver' attribute with the 'm' line.

  Note: An offerer compliant with [16] might not include 'floorctrl' and 'bfcpver' attributes in offers, in which cases the default values apply.

Once the answerer has sent the answer, the answerer:

- MUST, if the answerer is the 'active' endpoint, and if a TCP connection associated with the 'm' line is to be established (or re-established), initiate the establishing of the TCP connection; and
- MUST, if the answerer is the 'active' endpoint, and if an TLS/DTLS connection associated with the 'm' line is to be established (or re-established), initiate the establishing of the TLS/DTLS connection (by sending a ClientHello message).

If the answerer does not accept the 'm' line in the offer, it MUST assign a zero port value to the corresponding 'm' line in the answer. In addition, the answerer MUST NOT establish a TCP connection or a TLS/DTLS connection associated with the 'm' line.

13.3. Offerer Processing of the SDP Answer

When the offerer receives an answer, which contains an 'm' line with a non-zero port value, describing a BFCP stream, the offerer:

- MUST, if the offer is the 'active' endpoint, and if a TCP connection associated with the 'm' line is to be established (or re-established), initiate the establishing of the TCP connection; and
- MUST, if the offerer is the 'active' endpoint, and if an TLS/DTLS connection associated with the 'm' line is to be established (or re-established), initiate the establishing of the TLS/DTLS connection (by sending a ClientHello message).

Note: An answerer compliant with [16] might not include 'floorctrl' and 'bfcpver' attributes in answers, in which cases the default values apply.

If the 'm' line in the answer contains a zero port value, or if the offerer for some other reason does not accept the answer, the offerer MUST NOT establish a TCP connection or a TLS/DTLS connection associated with the 'm' line.

13.4. Modifying the Session

When an offerer sends an updated offer, in order to modify a previously established BFCP stream, it follows the procedures in Section 13.1, with the following exceptions:

- If the BFCP stream is carried on top of TCP, and if the offerer does not want to re-establish an existing TCP connection, the offerer MUST associate an SDP connection attribute with an 'existing' value, with the 'm' line; and

- If the offerer wants to disable a previously established BFCP stream, it MUST assign a zero port value to the 'm' line associated with the BFCP connection, following the procedures in [4].

14. Examples

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.

The following is an example of an offer sent by a conference server to a client.

```
m=application 50000 TCP/TLS/BFCP *
a=setup:actpass
a=connection:new
a=fingerprint:sha-256 \
     19:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04: \
     BB:05:2F:70:9F:04:A9:0E:05:E9:26:33:E8:70:88:A2
a=floorctrl:c-only s-only
a=confid:4321
a=userid:1234
a=floorid:1 mstrm:10
a=floorid:2 mstrm:11
a=bfcpver:1
m=audio 50002 RTP/AVP 0
a=label:10
m=video 50004 RTP/AVP 31
a=label:11
```

Note that due to RFC formatting conventions, this document splits SDP across lines whose content would exceed 72 characters. A backslash character marks where this line folding has taken place. This backslash and its trailing CRLF and whitespace would not appear in actual SDP content.

The following is the answer returned by the client.

```
m=application 9 TCP/TLS/BFCP *
a=setup:active
a=connection:new
a=fingerprint:sha-256 \
     6B:8B:F0:65:5F:78:E2:51:3B:AC:6F:F3:3F:46:1B:35: \
     DC:B8:5F:64:1A:24:C2:43:F0:A1:58:D0:A1:2C:19:08
a=floorctrl:c-only
a=bfcpver:1
m=audio 55000 RTP/AVP 0
m=video 55002 RTP/AVP 31
```

A similar example using unreliable transport and DTLS is shown below, where the offer is sent from a client.

```
m=application 50000 UDP/TLS/BFCP *
a=setup:actpass
a=dtls-id:abc3dl
a=fingerprint:sha-256 \
     19:E2:1C:3B:4B:9F:81:E6:B8:5C:F4:A5:A8:D8:73:04: \
     BB:05:2F:70:9F:04:A9:0E:05:E9:26:33:E8:70:88:A2
a=floorctrl:c-only s-only
a=confid:4321
a=userid:1234
a=floorid:1 mstrm:10
a=floorid:2 mstrm:11
a=bfcpver:2
m=audio 50002 RTP/AVP 0
a=label:10
m=video 50004 RTP/AVP 31
a=label:11
```

The following is the answer returned by the server.

```
m=application 55000 UDP/TLS/BFCP *
a=setup:active
a=dtls-id:abc3dl
a=fingerprint:sha-256 \
     6B:8B:F0:65:5F:78:E2:51:3B:AC:6F:F3:3F:46:1B:35: \
     DC:B8:5F:64:1A:24:C2:43:F0:A1:58:D0:A1:2C:19:08
a=floorctrl:s-only
a=confid:4321
a=userid:1234
a=floorid:1 mstrm:10
a=floorid:2 mstrm:11
a=bfcpver:2
m=audio 55002 RTP/AVP 0
m=video 55004 RTP/AVP 31
```

15. Security Considerations

The BFCP [7], SDP [10], and offer/answer [4] specifications discuss security issues related to BFCP, SDP, and offer/answer, respectively. In addition, [6] and [9] discuss security issues related to the establishment of TCP and TLS connections using an offer/answer model. Furthermore, when using DTLS over UDP, considerations for its use with RTP and RTCP are presented in [14]. The requirements for the offer/answer exchange, as listed in Section 5 of [14], MUST be followed.

An initial integrity-protected channel is REQUIRED for BFCP to exchange self-signed certificates between a client and the floor

control server. For session descriptions carried in SIP [3], S/MIME [5] is the natural choice to provide such a channel.

16. IANA Considerations

[Editorial note: The changes in Section 16.1 instruct the IANA to register the three new values TCP/DTLS/BFCP, UDP/BFCP and UDP/TLS/BFCP for the SDP 'proto' field. The new section Section 16.6 registers a new SDP "bfcpver" attribute. The rest is unchanged from [15].]

16.1. Registration of SDP 'proto' Values

The IANA has registered the following values for the SDP 'proto' field under the Session Description Protocol (SDP) Parameters registry:

Value	Reference
TCP/BFCP	[RFC XXXX]
TCP/DTLS/BFCP	[RFC XXXX]
TCP/TLS/BFCP	[RFC XXXX]
UDP/BFCP	[RFC XXXX]
UDP/TLS/BFCP	[RFC XXXX]

Table 3: Values for the SDP 'proto' field

16.2. Registration of the SDP 'floorctrl' Attribute

The IANA has registered the following SDP att-field under the Session Description Protocol (SDP) Parameters registry:

Contact name: iesg@ietf.org

Attribute name: floorctrl

Long-form attribute name: Floor Control

Type of attribute: Media level

Subject to charset: No

Purpose of attribute: The 'floorctrl' attribute is used to perform floor control server determination.

Allowed attribute values: 1*("c-only" / "s-only" / "c-s")

16.3. Registration of the SDP 'confid' Attribute

The IANA has registered the following SDP att-field under the Session Description Protocol (SDP) Parameters registry:

Contact name: iesg@ietf.org

Attribute name: confid

Long-form attribute name: Conference Identifier

Type of attribute: Media level

Subject to charset: No

Purpose of attribute: The 'confid' attribute carries the integer representation of a Conference ID.

Allowed attribute values: A token

16.4. Registration of the SDP 'userid' Attribute

The IANA has registered the following SDP att-field under the Session Description Protocol (SDP) Parameters registry:

Contact name: iesg@ietf.org

Attribute name: userid

Long-form attribute name: User Identifier

Type of attribute: Media level

Subject to charset: No

Purpose of attribute: The 'userid' attribute carries the integer representation of a User ID.

Allowed attribute values: A token

16.5. Registration of the SDP 'floorid' Attribute

The IANA has registered the following SDP att-field under the Session Description Protocol (SDP) Parameters registry:

Contact name: iesg@ietf.org

Attribute name: floorid

Long-form attribute name: Floor Identifier

Type of attribute: Media level

Subject to charset: No

Purpose of attribute: The 'floorid' attribute associates a floor with one or more media streams.

Allowed attribute values: Tokens

16.6. Registration of the SDP 'bfcpver' Attribute

The IANA has registered the following SDP att-field under the Session Description Protocol (SDP) Parameters registry:

Contact name: iesg@ietf.org

Attribute name: bfcpver

Long-form attribute name: BFCP Version

Type of attribute: Media level

Subject to charset: No

Purpose of attribute: The 'bfcpver' attribute lists supported BFCP versions.

Allowed attribute values: Tokens

17. Changes from RFC 4583

Following is the list of technical changes and other fixes from [16].

Main purpose of this work was to add signaling support necessary to support BFCP over unreliable transport, as described in [7], resulting in the following changes:

1. Fields in the 'm' line (Section 3):
   The section is re-written to remove reference to the exclusivity of TCP as a transport for BFCP streams. The proto field values TCP/DTLS/BFCP, UDP/BFCP and UDP/TLS/BFCP added.

2. Authentication (Section 11):
   In last paragraph, made clear that a TCP connection was described.

3. Security Considerations (Section 15):
   For the DTLS over UDP case, mention existing considerations and requirements for the offer/answer exchange in [14].

4. Registration of SDP 'proto' Values (Section 16.1):
   Register the three new values TCP/DTLS/BFCP, UDP/BFCP and UDP/TLS/BFCP in the SDP parameters registry.

5. BFCP Version Negotiation (Section 8):
   A new 'bfcpver' SDP media-level attribute is added in order to signal supported version number.

Clarification and bug fixes:

1. Errata ID: 712 (Section 4 and Section 13):
   Language clarification. Don't use terms like an SDP attribute is "used in an 'm' line", instead make clear that the attribute is a media-level attribute.

2. Fix typo in example (Section 14):
   Do not use 'm-stream' in the SDP example, use the correct 'mstrm' as specified in Section 14. Recommend interpreting 'm-stream' if it is received, since it is present in some implementations.

3. Assorted clarifications (Across the document):
   Language clarifications as a result of reviews. Also, the normative language where tightened where appropriate, i.e. changed from SHOULD strength to MUST in a number of places.

18. Acknowledgements

Joerg Ott, Keith Drage, Alan Johnston, Eric Rescorla, Roni Even, and Oscar Novo provided useful ideas for the original [16]. The authors also acknowledge contributions to the revision of BFCP for use over an unreliable transport from Geir Arne Sandbakken, Charles Eckel, Alan Ford, Eoin McLeod and Mark Thompson. Useful and important final reviews were done by Ali C. Begen, Mary Barnes and Charles Eckel. In the final stages, Roman Shpount made a considerable effort in adding proper ICE support and considerations.

19. References

19.1. Normative References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[2] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

[3] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <https://www.rfc-editor.org/info/rfc3261>.

[4] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <https://www.rfc-editor.org/info/rfc3264>.

[5] Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Certificate Handling", RFC 5750, DOI 10.17487/RFC5750, January 2010, <https://www.rfc-editor.org/info/rfc5750>.

[6] Yon, D. and G. Camarillo, "TCP-Based Media Transport in the Session Description Protocol (SDP)", RFC 4145, DOI 10.17487/RFC4145, September 2005, <https://www.rfc-editor.org/info/rfc4145>.

[7] Camarillo, G., Drage, K., Kristensen, T., Ott, J., and C. Eckel, "The Binary Floor Control Protocol (BFCP)", draft-ietf-bfcpbis-rfc4582bis-16 (work in progress), November 2015.

[8] Levin, O. and G. Camarillo, "The Session Description Protocol (SDP) Label Attribute", RFC 4574, DOI 10.17487/RFC4574, August 2006, <https://www.rfc-editor.org/info/rfc4574>.

[9] Lennox, J. and C. Holmberg, "Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP)", RFC 8122, DOI 10.17487/RFC8122, March 2017, <https://www.rfc-editor.org/info/rfc8122>.

[10] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <https://www.rfc-editor.org/info/rfc4566>.

[11] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://www.rfc-editor.org/info/rfc6347>.

[12] Lazzaro, J., "Framing Real-time Transport Protocol (RTP) and RTP Control Protocol (RTCP) Packets over Connection-Oriented Transport", RFC 4571, DOI 10.17487/RFC4571, July 2006, <https://www.rfc-editor.org/info/rfc4571>.

[13] Rosenberg, J., Keranen, A., Lowekamp, B., and A. Roach, "TCP Candidates with Interactive Connectivity Establishment (ICE)", RFC 6544, DOI 10.17487/RFC6544, March 2012, <https://www.rfc-editor.org/info/rfc6544>.

[14] Holmberg, C. and R. Shpount, "Session Description Protocol (SDP) Offer/Answer Considerations for Datagram Transport Layer Security (DTLS) and Transport Layer Security (TLS)", draft-ietf-mmusic-dtls-sdp-32 (work in progress), October 2017.

[15] Camarillo, G., Ott, J., and K. Drage, "The Binary Floor Control Protocol (BFCP)", RFC 4582, DOI 10.17487/RFC4582, November 2006, <https://www.rfc-editor.org/info/rfc4582>.

[16] Camarillo, G., "Session Description Protocol (SDP) Format for Binary Floor Control Protocol (BFCP) Streams", RFC 4583, DOI 10.17487/RFC4583, November 2006, <https://www.rfc-editor.org/info/rfc4583>.

[17] Keranen, A., Holmberg, C., and J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal", draft-ietf-ice-rfc5245bis-20 (work in progress), March 2018.

[18] Petit-Huguenin, M., Keranen, A., and S. Nandakumar, "Session Description Protocol (SDP) Offer/Answer procedures for Interactive Connectivity Establishment (ICE)", draft-ietf-mmusic-ice-sip-sdp-16 (work in progress), November 2017.

19.2. Informational References

[19] Lennox, J., Ott, J., and T. Schierl, "Source-Specific Media Attributes in the Session Description Protocol (SDP)", RFC 5576, DOI 10.17487/RFC5576, June 2009, <https://www.rfc-editor.org/info/rfc5576>.

[20] Holmberg, C., Alvestrand, H., and C. Jennings, "Negotiating Media Multiplexing Using the Session Description Protocol (SDP)", draft-ietf-mmusic-sdp-bundle-negotiation-49 (work in progress), March 2018.

[21] Nandakumar, S., "A Framework for SDP Attributes when Multiplexing", draft-ietf-mmusic-sdp-mux-attributes-17 (work in progress), February 2018.

Authors' Addresses

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: christer.holmberg@ericsson.com

Gonzalo Camarillo
Ericsson
Hirsalantie 11
FI-02420 Jorvas
Finland

Email: Gonzalo.Camarillo@ericsson.com

Tom Kristensen
Cisco
Philip Pedersens vei 1
NO-1366 Lysaker
Norway

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