Session Description Protocol (SDP) Format for Binary Floor Control Protocol (BFCP) Streams
draft-ietf-bfcpbis-rfc4583bis-12

Abstract

   This document specifies how to describe Binary Floor Control Protocol
   (BFCP) streams in Session Description Protocol (SDP) descriptions.
   User agents using the offer/answer model to establish BFCP streams
   use this format in their offers and answers.

   This document obsoletes RFC 4583. Changes from RFC 4583 are
   summarized in Section 14.

Status of this Memo

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

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

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

   This Internet-Draft will expire on March 14, 2016.

Table of Contents

1. Introduction

   As discussed in the BFCP (Binary Floor Control Protocol)
   specification [8], 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 [11], 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 [7]. Depending on the value of the 'setup' attribute
(discussed in Section 8.1), the port field contains the port to
which the remote endpoint will direct BFCP messages or is
irrelevant (i.e., the endpoint will initiate the connection
towards the remote endpoint) and should be set to a value of 9,
which is the discard port.

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 four values for the proto field: TCP/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. 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 [12], 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 SHOULD
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 Server Determination

When two endpoints establish a BFCP stream, they need to determine
which of them acts as a floor control server. In the most common
scenario, a client establishes a BFCP stream with a conference server
that acts as the floor control server. Floor control server
determination is straight forward because one endpoint can only act
as a client and the other can only act as a floor control server.

However, there are scenarios where both endpoints could act as a
floor control server. For example, in a two-party session that
involves an audio stream and a shared whiteboard, the endpoints need
to decide which party will be acting as the floor control server.

Furthermore, there are situations where both the offerer and the
answerer act as both clients and floor control servers in the same
session. For example, in a two-party session that involves an audio
stream and a shared whiteboard, one party acts as the floor control
server for the audio stream and the other acts as the floor control

server for the shared whiteboard.

4.1. SDP 'floorctrl' Attribute

This document defines the 'floorctrl' SDP media-level attribute to
perform floor control server determination. Its Augmented BNF syntax
[2] is:

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

The offerer includes this attribute to state all the roles it would
be willing to perform:

c-only: The offerer would be willing to act as a floor control
   client only.

s-only: The offerer would be 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.

If an SDP media description in an offer contains a 'floorctrl'
attribute, the answerer accepting that media MUST include a
'floorctrl' attribute in the corresponding media description of the
answer. The answerer includes this attribute to state which role the
answerer will perform. That is, the answerer chooses one of the
roles the offerer is willing to perform and generates an answer with
the corresponding role for the answerer. Table 1 shows the
corresponding roles for an answerer, depending on the offerer's role.

```
+---------+----------+
| Offerer | Answerer |
+---------+----------+
|  c-only |  s-only  |
|  s-only |  c-only  |
|   c-s   |    c-s   |
+---------+----------+
```

Table 1: Roles

The following are the descriptions of the roles when they are chosen
by an answerer:

   c-only: The answerer will act as a floor control client.
      Consequently, the offerer will act as a floor control server.

   s-only: The answerer will act as a floor control server.
      Consequently, the offerer will act as a floor control client.

   c-s: The answerer will act both as a floor control client and as a
      floor control server. Consequently, the offerer will also act
      both as a floor control client and as a floor control server.

   Endpoints that use the offer/answer model to establish BFCP
   connections MUST support the 'floorctrl' attribute. A floor control
   server acting as an offerer or as an answerer SHOULD include this
   attribute in its session descriptions.

   If the 'floorctrl' attribute is not used in an offer/answer exchange,
   by default the offerer and the answerer will act as a floor control
   client and as a floor control server, respectively.

   The following is an example of a 'floorctrl' attribute in an offer.
   When this attribute appears in an answer, it only carries one role:

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

5. SDP 'confid' and 'userid' Attributes

   This document defines the 'confid' and the 'userid' SDP media-level
   attributes. These attributes are used by a floor control server to
   provide a client with a conference ID and a user ID, respectively.
   Their Augmented BNF syntax [2] 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)

   The 'confid' and the 'userid' attributes carry the decimal integer
   representation of a conference ID and a user ID, respectively.

   The token-char and token elements are defined in [11] but included
   here to provide support for the implementor of this SDP feature.

   Endpoints that use the offer/answer model to establish BFCP

connections MUST support the 'confid' and the 'userid' attributes. A
floor control server acting as an offerer or as an answerer MUST
include these attributes in its session descriptions.

6. SDP 'floorid' Attribute

This document defines the 'floorid' SDP media-level attribute. This
attribute is used to provide an association between media streams and
floors. Its Augmented BNF syntax [2] is:

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

The 'floorid' attribute is used in the SDP media description for BFCP
media. It defines a floor identifier and, possibly, associates it
with one or more media streams. The token representing the floor ID
is the integer representation of the Floor ID to be used in BFCP.
The token representing the media stream is a pointer to the media
stream, which is identified by an SDP label attribute [9].

Endpoints that use the offer/answer model to establish BFCP
connections MUST support the 'floorid' and the 'label' attributes. A
floor control server acting as an offerer or as an answerer MUST
include these attributes in its session descriptions.

   Note: In [15] 'm-stream' was erroneously used in Section 11.
   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.

7. SDP 'bfcpver' Attribute

This document defines the 'bfcpver' SDP media-level attribute. This
attribute is used for BFCP version negotiation. Its Augmented BNF
syntax [2] is:

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

The 'bfcpver' attribute defines the list of the versions of BFCP
supported by the endpoint. Tokens representing versions MUST be
integers matching the "Version" field that would be presented in the
BFCP COMMON-HEADER [8]. The version of BFCP to be used will then be
confirmed with a BFCP-level Hello/HelloAck.

Endpoints that use the offer/answer model to establish BFCP

connections SHOULD support the 'bfcpver' attribute. A floor control
server acting as an offerer or as an answerer SHOULD include this
attribute in its session descriptions. However, endpoints that
support RFC XXXX, and not only the [15] subset, are REQUIRED to
support and, when acting as a floor control server, to use the
'bfcpver' attribute.

If a 'bfcpver' attribute is not present, default values are inferred
from the transport specified in the 'm' line (Section 3). In
accordance with definition of the Version field in [8], when used
over a reliable transport the default value is "1", and when used
over an unreliable transport the default value is "2".

8. BFCP Connection Management

BFCP connections can use TCP or UDP as the underlying transport.
BFCP entities exchanging BFCP messages over UDP direct the BFCP
messages to the peer side connection address and port provided in the
SDP 'm' line. TCP connection management is more complicated and is
described below.

8.1. TCP Connection Management

The management of the TCP connection used to transport BFCP is
performed using the 'setup' and 'connection' attributes, as defined
in [7].

The 'setup' attribute indicates which of the endpoints (client or
floor control server) initiates the TCP connection. The 'connection'
attribute handles TCP connection reestablishment.

The BFCP specification [8] describes a number of situations when the
TCP connection between a client and the floor control server needs to
be reestablished. However, that specification does not describe the
reestablishment process because this process depends on how the
connection was established in the first place. BFCP entities using
the offer/answer model follow the following rules.

When the existing TCP connection is closed and reestablished
following the rules in [8], the client MUST generate an offer towards
the floor control server in order to reestablish the connection. If
a TCP connection cannot deliver a BFCP message and times out, the
entity that attempted to send the message (i.e., the one that
detected the TCP timeout) MUST generate an offer in order to
reestablish the TCP connection.

Endpoints that use the offer/answer model to establish TCP

connections MUST support the 'setup' and 'connection' attributes.

9. **Authentication**

   When a BFCP connection is established using the offer/answer model,
   it is assumed that the offerer and the answerer authenticate each
   other using some mechanism. TLS/DTLS is the preferred mechanism, but
   other mechanisms are possible and 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.

   When SIP is used to perform an offer/answer exchange, the initial
   mutual authentication takes place at the SIP level. Additionally,
   SIP uses S/MIME [6] 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.

   BFCP clients and floor control servers follow the rules in [10]
   regarding certificate choice and presentation. This implies that
   unless a 'fingerprint' attribute is included in the session
   description, the certificate provided at the TLS-/DTLS-level MUST
   either be directly signed by one of the other party's trust anchors
   or be validated using a certification path that terminates at one of
   the other party's trust anchors [5]. Endpoints that use the offer/
   answer model to establish BFCP connections MUST support the
   'fingerprint' attribute and MUST include it in their session
   descriptions.

   When TLS is used with TCP, once the underlying connection is
   established, the answerer which may be the 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 [13] 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 [15] for TCP to
retain backwards compatibility. When using UDP, the procedure
above was preferred since it adheres to [13] as used for DTLS-
SRTP, it does not overload offer/answer semantics, and it works
for offerless INVITE in scenarios with B2BUAs.

10. SDP Offer/Answer Procedures

This section defines the SDP offer/answer [4] procedures for
negotiating and establishing a BFCP connection.

If the 'm' line 'proto' value is 'TCP/TLS/BFCP' or 'UDP/TLS/BFCP',
each endpoint MUST provide a certificate fingerprint, using the SDP
'fingerprint' attribute [7], if the endpoint supports, and is willing
to use, a cipher suite with an associated certificate.

The authentication certificates are interpreted and validated as
defined in [10]. Self-signed certificates can be used securely,
provided that the integrity of the SDP description is assured as
defined in [10].

Note: The procedures apply to a specific 'm' line describing a
BFCP connection. If an offer or answer contains multiple 'm'
lines describing BFCP connections, the procedures are applied
separately to each 'm' line.

10.1. Generating the Initial SDP Offer

When the offerer creates an initial offer, the offerer:

o MUST, if the 'm' line proto value is 'TCP/BFCP', 'TCP/TLS/BFCP' or
   'UDP/TLS/BFCP', associate an SDP setup attribute, with an
   'actpass' value, with the 'm' line;

o MUST, if the 'm' line proto value is 'TCP/BFCP' or 'TCP/TLS/BFCP',
   associate an SDP 'connection' attribute, with a 'new' value, with
   the 'm' line; and

In addition, if the offerer acts as the floor control server, the
offerer:

o SHOULD associate an SDP 'floorctrl' attribute defined in
   Section 4.1, with the 'm' line;

o MUST associate an SDP 'confid' attribute defined in [Section 5](), with the 'm' line;

o MUST associate an SDP 'userid' attribute defined in [Section 5](), with the 'm' line;

o MUST associate an SDP 'floorid' attribute defined in [Section 6](), with the 'm' line;

o MUST associate an SDP 'label' attribute as described in [Section 6](), with the 'm' line; and

o SHOULD, if it supports only the [RFC 4583]() subset and MUST, if it supports RFC XXXX associate an SDP 'bfcpver' attribute defined in [Section 7](), with the 'm' line.

10.2. Generating the SDP Answer

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

o MUST insert a corresponding 'm' line in the answer, with an identical 'm' line proto value [4]; and

o MUST, if the 'm' line proto value is 'TCP/BFCP', 'TCP/TLS/BFCP' or 'UDP/TLS/BFCP', associate an SDP setup attribute, with an 'active' or 'passive' value, with the 'm' line;

In addition, if the answerer acts as the floor control server, the answerer:

o MUST, if the offer contains a 'floorctrl' attribute or else it SHOULD associate an SDP 'floorctrl' attribute defined in [Section 4.1](), with the 'm' line;

o MUST associate an SDP 'confid' attribute defined in [Section 5](), with the 'm' line;

o MUST associate an SDP 'userid' attribute defined in [Section 5](), with the 'm' line;

o MUST associate an SDP 'floorid' attribute defined in [Section 6](), with the 'm' line; and

o MUST associate an SDP 'label' attribute as described in [Section 6](), with the 'm' line.

o SHOULD, if it supports only the RFC 4583 subset and MUST, if it
   supports RFC XXXX associate an SDP 'bfcpver' attribute defined in
   Section 7, with the 'm' line.

Once the answerer has sent the answer, the answerer:

o 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

o 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.

10.3. 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 connection, the offerer:

o 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

o 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).

If the 'm' line in the answer contains a zero port value, the offerer
MUST NOT establish a TCP connection or a TLS/DTLS connection
associated with the 'm' line.

10.4. Modifying the Session

When an offerer sends an updated offer, in order to modify a
previously established BFCP connection, it follows the procedures in
Section 10.1, with the following exceptions:

   o If the BFCP connection is carried on top of TCP, and unless the
      offerer wants to re-establish an existing TCP connection, the
      offerer MUST associate an SDP connection attribute, with an
      'existing' value, with the 'm' line; and

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

10.5. DTLS Role Determination

   If the 'm' line proto value is 'UDP/TLS/BFCP', the 'active/passive'
   status is used to determine the TLS roles. Following the procedures
   in [10], the 'active' endpoint will take the TLS client role.

   Once a DTLS connection has been established, if the 'active/passive'
   status of the endpoints change during a session, a new DTLS
   connection MUST be established. Therefore, endpoints SHOULD NOT
   change the 'active/passive' status in subsequent offers and answers,
   unless they want to establish a new DTLS connection.

   The conditions under which endpoints MUST establish a new DTLS
   connection are as the same defined for DTLS-SRTP in [13].

11. 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:passive
   a=connection:new
   a=fingerprint:SHA-1 \
        4A:AD:B9:B1:3F:82:18:3B:54:02:12:DF:3E:5D:49:6B:19:E5:7C:AB
   a=floorctrl: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-1 \
        3D:B4:7B:E3:CC:FC:0D:1B:5D:31:33:9E:48:9B:67:FE:68:40:E8:21
   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=fingerprint:SHA-1 \
     4A:AD:B9:B1:3F:82:18:3B:54:02:12:DF:3E:5D:49:6B:19:E5:7C:AB
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=fingerprint:SHA-1 \
     3D:B4:7B:E3:CC:FC:0D:1B:5D:31:33:9E:48:9B:67:FE:68:40:E8:21
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
```

12. Security Considerations

The BFCP [8], SDP [11], and offer/answer [4] specifications discuss
security issues related to BFCP, SDP, and offer/answer, respectively.
In addition, [7] and [10] 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 [13]. The requirements for the
offer/answer exchange, as listed in Section 5 of [13], 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
[6] is the natural choice to provide such a channel.

13. IANA Considerations

   [Editorial note: The changes in Section 13.1 instruct the IANA to
   register the two new values UDP/BFCP and UDP/TLS/BFCP for the SDP
   'proto' field. The new section Section 13.6 registers a new SDP
   "bfcpver" attribute. The rest is unchanged from [14].]

13.1. 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/TLS/BFCP | [RFC XXXX] |
               | UDP/BFCP     | [RFC XXXX] |
               | UDP/TLS/BFCP | [RFC XXXX] |
               +--------------+------------+
```

               Table 2: Values for the SDP 'proto' field

13.2. 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: Gonzalo.Camarillo@ericsson.com

   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")

13.3. 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: Gonzalo.Camarillo@ericsson.com

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

13.4. 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: Gonzalo.Camarillo@ericsson.com

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

13.5. 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: Gonzalo.Camarillo@ericsson.com

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

13.6. 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: Gonzalo.Camarillo@ericsson.com

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

14. Changes from RFC 4583

Following is the list of technical changes and other fixes from [15].

Main purpose of this work was to add signaling support necessary to
support BFCP over unreliable transport, as described in [8],
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
    UDP/BFCP and UDP/TLS/BFCP added.

2. Authentication (Section 9):
    In last paragraph, made clear that a TCP connection was
    described.

3. Security Considerations (Section 12):
    For the DTLS over UDP case, mention existing considerations and
    requirements for the offer/answer exchange in [13].

4. Registration of SDP 'proto' Values (Section 13.1):
    Register the two new values UDP/BFCP and UDP/TLS/BFCP in the SDP
    parameters registry.

5. BFCP Version Negotiation (Section 7):
    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 6):
    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 11):
    Do not use 'm-stream' in the SDP example, use the correct 'mstrm'
    as specified in Section 11. 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.

15. Acknowledgements

   Joerg Ott, Keith Drage, Alan Johnston, Eric Rescorla, Roni Even, and
   Oscar Novo provided useful ideas for the original [15]. 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.

16. Normative References

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

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

[3]   Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A.,
        Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP:
        Session Initiation Protocol", RFC 3261, June 2002.

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

[5]   Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley,
        R., and W. Polk, "Internet X.509 Public Key Infrastructure
        Certificate and Certificate Revocation List (CRL) Profile",
        RFC 5280, May 2008.

[6]   Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet Mail
        Extensions (S/MIME) Version 3.2 Certificate Handling",
        RFC 5750, January 2010.

[7]   Yon, D. and G. Camarillo, "TCP-Based Media Transport in the
        Session Description Protocol (SDP)", RFC 4145, September 2005.

[8]   Camarillo, G., Drage, K., Kristensen, T., Ott, J., and C.
        Eckel, "The Binary Floor Control Protocol (BFCP)",
        draft-ietf-bfcpbis-rfc4582bis-13 (work in progress),
        February 2015.

[9]   Levin, O. and G. Camarillo, "The Session Description Protocol
        (SDP) Label Attribute", RFC 4574, August 2006.

[10]   Lennox, J., "Connection-Oriented Media Transport over the
        Transport Layer Security (TLS) Protocol in the Session
        Description Protocol (SDP)", RFC 4572, July 2006.

[11]   Handley, M., Jacobson, V., and C. Perkins, "SDP: Session
        Description Protocol", RFC 4566, July 2006.

[12]   Rescorla, E. and N. Modadugu, "Datagram Transport Layer
        Security Version 1.2", RFC 6347, January 2012.

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

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

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

Authors' Addresses

   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

   Paul E. Jones
   Cisco
   7025 Kit Creek Rd.
   Research Triangle Park, NC 27709
   USA

   Email: paulej@packetizer.com