Network Working Group
Request for Comments: 4583
Category: Standards Track

G. Camarillo
Ericsson

November 2006

Session Description Protocol (SDP) Format for Binary Floor Control Protocol (BFCP) Streams

Status of This Memo

Copyright Notice

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.

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. These data include 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 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

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" 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> <transport> <fmt> ...
```

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

The port field is set following the rules in [7]. Depending on the value of the 'setup' attribute (discussed in Section 7), the port field contains the port to which the remote endpoint will initiate its TCP connection 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. Since BFCP only runs on top of TCP, the port is always a TCP port. A port field value of zero has the standard SDP meaning (i.e., rejection of the media stream).

We define two new values for the transport field: 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 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.

We define the 'floorctrl' SDP media-level attribute to perform floor control 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 'm' line in an offer contains a 'floorctrl' attribute, the answerer MUST include one in the corresponding 'm' line in 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. The 'confid' and 'userid' SDP Attributes

We define 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
```

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

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 SHOULD include these attributes in its session descriptions.

6. Association between Streams and Floors

We define the 'floorid' SDP media-level attribute. Its Augmented BNF syntax [2] is:

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

The 'floorid' attribute is used in BFCP 'm' lines. 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 SHOULD include these attributes in its session descriptions.

7. 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 reset following the rules in [8], the client SHOULD 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) SHOULD generate an offer in order to reestablish the TCP connection.

Endpoints that use the offer/answer model to establish BFCP connections MUST support the 'setup' and 'connection' attributes.

8. 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. 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 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-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 SHOULD include it in their session descriptions.

When TLS is used, once the underlaying TCP connection is established, the answerer acts as the TLS server regardless of its role (passive or active) in the TCP establishment procedure.

9. 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 m-stream:10
a=floorid:2 m-stream:11
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
m=audio 55000 RTP/AVP 0
m=video 55002 RTP/AVP 31
```

10. 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.

BFCP assumes that an initial integrity-protected channel is used 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.

11. IANA Considerations

11.1. Registration of the 'TCP/BFCP' and 'TCP/TLS/BFCP' SDP 'proto' Values

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

Value	Reference
TCP/BFCP	RFC4583
TCP/TLS/BFCP	RFC4583

Table 2: Values for the SDP 'proto' field

11.2. 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")

11.3. 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

11.4. Registration of the SDP 'userid' Attribute

This section instructs the IANA to register 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

11.5. Registration of the SDP 'floorid' Attribute

This section instructs the IANA to register 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

12. Acknowledgements

Joerg Ott, Keith Drage, Alan Johnston, Eric Rescorla, Roni Even, and Oscar Novo provided useful ideas for this document.

13. Normative References

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

[2] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.

[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] Housley, R., Polk, W., Ford, W., and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002.

[6] Ramsdell, B., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Certificate Handling", RFC 3850, July 2004.

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

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

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

Author's Address

Gonzalo Camarillo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

EMail: Gonzalo.Camarillo@ericsson.com

Full Copyright Statement

Intellectual Property

Acknowledgement

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