Connection-Oriented Media Transport in the Session Description Protocol (SDP)
draft-ietf-mmusic-sdp-comedia-07.txt

Abstract

This document describes how to express media transport over connection-oriented protocols using the Session Description Protocol (SDP). It defines the SDP TCP protocol identifier, the SDP setup attribute, which describes the connection setup procedure, and the SDP connid attribute, which provides a connection identifier.
Table of Contents

1. Introduction ............................................. 3
2. Terminology ............................................. 3
3. Protocol Identifier ..................................... 3
4. Setup Attribute ......................................... 4
   4.1 The Setup Attribute in the Offer/answer Model ........ 4
5. The Connid Attribute .................................... 5
   5.1 Offerer Behaviour ..................................... 6
   5.2 Answerer Behaviour ..................................... 6
6. Connection Management ................................... 7
7. Examples ................................................ 8
   7.1 Passive/Active ......................................... 8
   7.2 Passive/Active with Connection Reestablishment ........ 8
   7.3 Actpass/Passive ......................................... 9
8. Security Considerations .................................. 9
9. IANA Considerations ...................................... 10
10. Acknowledgements ....................................... 10
11. References .............................................. 10
   11.1 Normative References .................................. 10
   11.2 Informational References ............................. 10
   Authors’ Addresses ....................................... 11
   Intellectual Property and Copyright Statements .......... 12
1. Introduction

The Session Description Protocol [3] provides a general-purpose format for describing multimedia sessions in announcements or invitations. SDP uses an entirely textual data format (the US-ASCII subset of UTF-8 [5]) to maximize portability among transports. SDP does not define a protocol, but only the syntax to describe a multimedia session with sufficient information to participate in that session. Session descriptions may be sent using arbitrary existing application protocols for transport (e.g., SAP [9], SIP [10], RTSP [6], email, HTTP [8], etc.).

SDP [3] defines two protocol identifiers: RTP/AVP and UDP, both of which represent unreliable connectionless protocols. While these transports are appropriate choices for multimedia streams, there are applications for which connection-oriented transports, such as TCP, are more appropriate. We define a new protocol identifier, TCP, to describe TCP connections in SDP.

Connection-oriented protocols introduce two new factors when describing a session: how and when should endpoints perform the connection setup procedure. We define two new attributes to describe connection setups: setup and connid.

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 [2] and indicate requirement levels for compliant implementations.

3. Protocol Identifier

The following is the ABNF for an m= line, as specified by RFC 2327 [3].

```plaintext
media-field = "m=" media space port ["/" integer]
             space proto 1*(space fmt) CRLF
```

We define a new value for the proto field: TCP.

The TCP protocol identifier is similar to the UDP protocol identifier in that it only describes the transport protocol, and not the upper-layer protocol. An m= line that specifies "TCP" MUST further qualify the application-layer protocol using an fmt identifier. Media lines with the TCP protocol identifier are carried over TCP [1].
It is RECOMMENDED that documents defining new SDP protocol identifiers that involve extra protocol layers between TCP and the media itself (e.g., TLS [7] over TCP) start with the string "TCP/" (e.g., TCP/TLS).

The following sections define the setup and the connid attributes. While they are applicable to m= lines that use the TCP protocol identifier, they are not limited to them. These attributes SHOULD be used in any m= line which uses a connection-oriented transport protocol, even if the protocol identifier of the m= line is not TCP.

4. 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 is charset-independent and can be a session-level or a media-level attribute. The following is the ABNF of the setup attribute:

```
setup-attr = "a=setup:" role
role       = "active" / "passive" / "actpass"
```

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.

4.1 The Setup Attribute in the Offer/answer Model

The offer/answer model, defined in RFC 3264 [4], provides endpoints with a means to obtain shared view of a session. Some session parameters are negotiated (e.g., codecs to use), while others are simply communicated from one endpoint to the other (e.g., IP addresses). The value of the setup attribute falls into the first category. That is, both endpoints negotiate its value using the offer/answer model.

The negotiation of the value of the setup attribute takes places as follows. The offerer states which role or roles is willing to perform and the answerer, taking the offerer’s willingness into consideration, chooses which roles both endpoints will actually perform during connection establishment. The following are the values that the setup attribute can take in an offer/answer exchange:

<table>
<thead>
<tr>
<th>Offer</th>
<th>Answer</th>
</tr>
</thead>
</table>
The value active indicates that the endpoint SHOULD initiate a connection to the port number on the m= line of the other endpoint. The port number on its own m= line is irrelevant, and the opposite endpoint MUST NOT attempt to initiate a connection to the port number specified there. Nevertheless, since the m= line must contain a valid port number, the endpoint specifying using the value active SHOULD specify a port number of 9 (the discard port) on its m= line. The endpoint MUST NOT specify a port number of zero, except to denote an m= line that has been or is being refused.

The value passive indicates that the endpoint SHOULD be ready to accept a connection on the port number specified in the m= line.

The value actpass indicates that the offerer can either initiate a connection to the port number on the m= line in the answer or accept a connection on the port number specified in the m= line in the offer. That is, the offerer has no preference as to whether it accepts or initiates the connection and, so, is letting the answerer choose.

The default value of the setup attribute in an offer/answer exchange is active in the offer and passive in the answer.

5. The Connid Attribute

The preceding description of the setup attribute has been in the context of using SDP to initiate a session. Still, 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 media connection unchanged, or is a request to make a new media connection. The media-level connid attribute, which is charset-independent, is used to disambiguate these two scenarios. The following is the ABNF of the connid attribute:

```
connid                 = "a=connid:" connection-identifier
connection-identifier  = token
```

The connid attribute provides an identifier for the transport-layer connection used by the m= line. Connid values are meaningful in the
context of a particular m= line. So, different m= lines in the same session description MAY have the same connid value.

5.1 Offerer Behaviour

Offerers and answerers use the connid 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.

When an offerer generates an m= line which uses a connection-oriented transport, it SHOULD provide such an m= line with a connection identifier using a connid attribute, unless the application using the m= line has other means to deal with connection reestablishment. The connid attribute in an initial offer (i.e., no transport connection has been established yet) can take any value. This value identifies the initial connection that the endpoints will attempt to establish.

After the initial offer/answer exchange, any of the endpoints can generate a new offer to change some characteristics of the session (e.g., the direction attribute). If such an offerer wants to continue using the previously-established transport-layer connection for the m= line, the offerer MUST use the same connid value for the m= line.

If, on the other hand, the offerer wants to establish a new transport-layer connection for the m= line, it MUST use a new connid value. This new connid value MUST be different from the current connid value in use and SHOULD be different than any connid value used previously in the same m= line.

The connid value in an offer is only compared with the connid value currently in use. So, having a connid value different than the one in use is enough to trigger the establishment of a new connection. Still, we recommend to use a value different than all the previous ones used in the m= line to make debugging easier. Note that, according to the rules in this section, an offer that changes the transport address (IP address plus port number) of an m= line will have a new connid value for this m= line.

5.2 Answerer Behaviour

The connid value for an m= line is negotiated using the offer/answer model. The resulting connid value after an offer/answer exchange is the connid value in the answer.

For an m= line, if the offer contains a new connid value (i.e., different from the one in use) the answerer MUST use this value in the answer. If the offer contains the connid value in use and the answerer wishes to continue using the existing transport-layer connection, the answerer MUST use this connid value in the answer. If
the offer contains the connid value in use but the answerer wishes to establish a new transport-layer connection, the answerer MUST use a new connid value in the answer.

If the connid value for an m= line resulting from an offer/answer exchange is different than the connid in use so far, the endpoints SHOULD establish a new transport-layer connection as indicated by the setup attribute. If a previous connection is still up, the endpoint responsible for establishing the new connection performing the active role SHOULD close it as soon as the offer/answer exchange is completed.

If the connid value for an m= line resulting from an offer/answer exchange is the same as the connid in use so far, the endpoints SHOULD continue using the existing connection.

In the past, it was proposed to use the presence of a media-level SDP attribute as a flag to indicate that a new connection needed to be established. We chose not to follow the flag approach because an offerer whose intent was to signal "no changes" in a session would need to issue a different offer than the previous one (i.e., it would need to remove the flag from the m= line). By using the connid attribute instead, an offerer signals "no changes" in a session by issuing an identical offer to the one in use.

6. Connection Management

An endpoint that according to an offer/answer exchange is supposed to initiate a new connection SHOULD initiate it as soon as the offer/answer exchange is completed, even if the endpoint does not intend to immediately begin sending media to the remote endpoint. This allows media to flow from the remote endpoint if needed.

Typically, endpoints do not close the connection until the session has expired, been explicitly terminated, or a new connid value has been provided for the m= line. Additionally, specific applications can describe further scenarios where an end-point may close a given connection. In case the session is explicitly terminated by one of the endpoints (e.g., the endpoint sends a SIP [10] BYE), the end point terminating the session is responsible for closing the transport-connection.

If an endpoint determines that the transport-connection for an m= line has been closed and it should be reestablished, it SHOULD perform a new offer/answer exchange using a new connid value for this m= line.
Note that the SDP direction attribute (e.g., a=sendonly) deals with the media sent over the transport-connection, but has no impact on the transport-connection itself.

7. Examples

The following examples show the most common usage of the setup attribute combined with TCP-based media descriptions. 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 (including c= lines).

7.1 Passive/Active

An offerer at 192.0.2.2 signals its availability for a T.38 fax session at port 54111:

m=image 54111 TCP t38
  c=IN IP4 192.0.2.2
  a=setup:passive
  a=connid:1

An answerer at 192.0.2.1 receiving this offer responds with the following answer:

  c=IN IP4 192.0.2.1
  m=image 9 TCP t38
  a=setup:active
  a=connid:1

The endpoint at 192.0.2.1 then initiates the TCP connection to port 54111 at 192.0.2.2.

7.2 Passive/Active with Connection Reestablishment

Continuing the preceding example, consider the scenario where the TCP connection fails and the endpoints wish to reestablish the connection for the session. The endpoint at 192.0.2.2 signals this intent with the following SDP:

m=image 54111 TCP t38
  c=IN IP4 192.0.2.2
  a=setup:passive
  a=connid:2
The new connid value informs the endpoint at 192.0.2.1 that this SDP represents the intent to establish a new connection for media transport, rather than continuing with the original connection. If 192.0.2.1 agrees to continue the session using a new connection, it responds with:

```
m=image 9 TCP t38
c=IN IP4 192.0.2.1
a=setup:active
a=connid:2
```

### 7.3 Actpass/Passive

In another example, an offerer at 192.0.2.2 signals its availability for a T.38 fax session at TCP port 54111. Additionally, this offerer is also willing to set up the media stream by initiating the TCP connection:

```
m=image 54111 TCP t38
c=IN IP4 192.0.2.2
a=setup:actpass
a=connid:3
```

The endpoint at 192.0.2.1 responds with the following description:

```
m=image 54321 TCP t38
c=IN IP4 192.0.2.1
a=setup:passive
a=connid:3
```

This will cause the offerer (at 192.0.2.2) to initiate a connection to port 54321 at 192.0.2.1.

### 8. Security Considerations


An attacker may attempt to modify the values of the connid attributes to have endpoints reestablish connections unnecessarily. So, it is STRONGLY RECOMMENDED that integrity protection be applied to the SDP session descriptions. For session descriptions carried in SIP [10], S/MIME is the natural choice to provide such end-to-end integrity protection, as described in RFC 3261 [10]. Other applications MAY use
a different form of integrity protection.

9. IANA Considerations

This document defines two session and media level SDP attributes: setup and connid. Their formats are defined in Section 4 and Section 5 respectively. These two attributes should be registered by the IANA on

http://www.iana.org/assignments/sdp-parameters

under "att-field (both session and media level)".

This document defines a proto values: TCP. Its format is defined in Section 3. This proto value should be registered by the IANA on

http://www.iana.org/assignments/sdp-parameters

under "proto".

10. Acknowledgements

Jonathan Rosenberg, Rohan Mahy, Anders Kristensen, Joerg Ott, Paul Kyzivat, Robert Fairlie-Cuninghame, Colin Perkins, and Christer Holmberg provided valuable insights and contributions.

11. References

11.1 Normative References


11.2 Informational References

Protocol (RTSP)", RFC 2326, April 1998.


Authors’ Addresses

David Yon
Dialout.Net, Inc
One Indian Head Plaza
Nashua, NH 03060
USA

EMail: yon@dialout.net

Gonzalo Camarillo
Ericsson
Hirsialantie 11
Jorvas 02420
Finland

EMail: Gonzalo.Camarillo@ericsson.com
Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the IETF's procedures with respect to rights in IETF Documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The Internet Society (2004). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

Acknowledgment

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