

Connection-Oriented Media Transport in SDP
<[draft-ietf-mmusic-sdp-comedia-05.txt](#)>

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at:
<http://www.ietf.org/ietf/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at:
<http://www.ietf.org/shadow.html>.

Copyright (C) The Internet Society (2002). All Rights Reserved.

Abstract

This document describes how to express media transport over connection-oriented protocols using the Session Description Protocol (SDP). It defines two new protocol identifiers: TCP and TLS. It also defines the syntax and semantics for an SDP "direction" attribute that describes the connection setup procedure.

1 Introduction

The Session Description Protocol [[SDP](#)] 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](#)]) to maximize portability among transports. SDP does not define a protocol, but only the syntax to describe a multimedia session with sufficient information to discover and participate in that session. Session descriptions may be sent using arbitrary existing application protocols for transport (e.g., SAP, SIP, RTSP, email, HTTP, etc.).

[SDP] describes two protocol identifiers: RTP/AVP and UDP, both of which are unreliable, connectionless protocols, an appropriate choice for multimedia streams. There are, however, applications for which the connection-oriented transports such as TCP are more appropriate, but [[SDP](#)] provides no way to describe a session that uses protocols other than RTP or UDP.

Connection-oriented protocols introduce a new factor when describing a session: not only must it be possible to express that a protocol will be based on this protocol, but it must also describe the connection setup procedure. This memo defines two new protocol identifiers, TCP and TLS, along with the syntax and semantics of the a=direction and a=reconnect attributes.

2 Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in [RFC 2119](#) [7] and indicate requirement levels for compliant implementations.

3 Protocol Identifiers

The m= line in [[SDP](#)] is where an endpoint specifies the protocol used for the media in the session. See the "Media Announcements" section of [[SDP](#)] for a discussion on protocol identifiers.

3.1 TCP

The TCP protocol identifier is similar to the UDP protocol identifier in that it only describes the transport protocol without any connotation as to the upper-layer protocol. An m= line that specifies "TCP" MUST further qualify the protocol using a fmt

identifier (see [[SDP](#)] [Appendix B](#)).

[3.2](#) TLS

The TLS protocol identifier specifies that the session will use the Transport Layer Security protocol [[TLS](#)] with an implied transport protocol of TCP. To describe a media session that uses TLS over TCP, the protocol identifier "TLS" must be specified in the m= line.

Yon INTERNET-DRAFT - Expires September 2003 2

INTERNET-DRAFT Connection-Oriented Media in SDP March 2003

An m= line that specifies TLS MUST further qualify the protocol using a fmt identifier.

[4](#) Direction Attribute

An important attribute of connection-oriented protocols is the setup procedure. One endpoint needs to initiate the connection and the other endpoint needs to accept the connection. The direction attribute is used to describe these roles, and the syntax is as follows:

a=direction:<role>

The <role> is one of the following:

passive: The endpoint will accept an incoming connection.

active: The endpoint will initiate an outgoing connection.

both: The endpoint will both accept an incoming connection
 and will initiate an outgoing connection.

[4.1](#) Semantics of direction:passive

By specifying direction:passive, the endpoint indicates that the port number specified in the m= line is available to accept a connection from the other endpoint.

[4.2](#) Semantics of direction:active

By specifying direction:active, the endpoint indicates that it will 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 direction: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, as that carries other semantics in [[SDP](#)]. The following SDP fragment shows an example of direction:active:

```
c=IN IP4 10.1.1.1
m=image 9 TCP t38
a=direction:active IN IP4
```

[4.3](#) Semantics of direction:both

By specifying direction:both, the endpoint indicates that it will both accept a TCP connection on the port number of its own m= line, and that it will also initiate a connection to the port number on the m= line of the other endpoint.

| | | |
|----------------|---|------------|
| Yon | INTERNET-DRAFT - Expires September 2003 | 3 |
| INTERNET-DRAFT | Connection-Oriented Media in SDP | March 2003 |

Since this attribute describes behavior that is similar to connectionless media descriptions in [[SDP](#)], it is the default value for the direction attribute and is therefore optional.

Endpoints may choose to specify direction:both for one or more of the following reasons:

- 1) The endpoint has no preference as to whether it accepts or initiates the connection, and therefore is offering the remote endpoint a choice of connection setup procedures.
- 2) The endpoints intend to use a single connection to transport the media, but it is not known whether firewall issues will prevent either endpoint from initiating or accepting the connection. Therefore both endpoints will attempt to initiate a connection in hopes that at least one will succeed.

If one endpoint specifies either direction:active or direction:passive and the other specifies direction:both, both endpoints MUST behave as if the latter had specified the inverse direction of the former. For example, specifying direction:both when the other endpoint specifies direction:active SHALL cause both endpoints to behave as if the former had specified direction:passive. Conversely, specifying direction:both when the other endpoint specifies direction:passive SHALL cause both endpoints to behave as if the former had specified direction:active.

If both endpoints specify direction:both then each endpoint MUST initiate a connection to the port number specified on the m= line of the opposite endpoint. There is one exception to this requirement: if an endpoint receives the incoming connection from the opposite endpoint prior to initiating its own outbound connection, then that

endpoint MAY use that connection rather than attempt to make an outbound connection to the opposite endpoint.

If only one connection succeeds, then that connection will be used to carry the media. Once it has transmitted data on this connection, the initiating endpoint MUST NOT perform another connection attempt to the accepting endpoint. This allows the accepting endpoint to release or recycle the listening port for another session once it has received data from the initiating endpoint.

If both connections succeed, the following rules SHALL apply:

- a) Each endpoint MUST accept data from either connection.
- b) Once an endpoint has transmitted data to one of the connections, it MUST use that connection exclusively for transmission.
- c) Once an endpoint has transmitted AND received data, if one of the connections is determined to be idle, the endpoint SHOULD close the idle connection.

| | | |
|----------------|---|------------|
| Yon | INTERNET-DRAFT - Expires September 2003 | 4 |
| INTERNET-DRAFT | Connection-Oriented Media in SDP | March 2003 |

4.4 Optimizing direction:both

As discussed in the previous section, there is the possibility that two connections will be created when only one is needed. While rules in the previous section accommodate the closing of an idle connection, they do not prevent a race condition where the endpoints simultaneously start sending data on opposite connections thereby causing two connections to be used where one would have sufficed. While it is not possible to entirely eliminate this race condition, it is in the endpoints' interest to minimize its occurrence. Therefore, when a session is negotiated through interactive exchange of SDP between endpoints (as in the case of SIP) AND the result of the negotiation is that each endpoint specifies direction:both, it is RECOMMENDED that the endpoints use the following guidelines:

- a) There comes a point during the exchange of SDP where one endpoint is prepared to send the final message that will complete the negotiation and allow the session to begin. For the purposes of this discussion, the endpoint that will send this final message will be called the Initiator, and the endpoint that will receive this message will be called the Acceptor.
- b) The Initiator, upon receiving sufficient information to initiate a connection, MUST attempt to connect to the Acceptor as soon as possible.

- c) In order to lower the likelihood that the Acceptor will also attempt to initiate a connection, the Initiator SHOULD incorporate a short delay between initiating the connection and sending the final SDP to the Acceptor.
- d) The delay time chosen by the Initiator MUST NOT introduce an unacceptable session setup delay should the connection to the Acceptor not succeed.

4.5 Bidirectional versus Unidirectional Media

In traditional SDP transport types the flow is unidirectional. If the intent is for media to flow in both directions, both endpoints must specify SDP that describes where to deliver the media and what media type(s) to use. For example, if only Endpoint A presents SDP then media can only flow towards Endpoint A, as Endpoint B has not specified where and how to send media to it.

Because most connection-oriented media is inherently bi-directional, endpoints may encounter a situation where only one side presented SDP yet there is now a network path that can carry media in either direction. In keeping with traditional SDP semantics, an endpoint MUST NOT send data to the other endpoint unless it has specified SDP information describing the type of media it can accept.

It is, however, perfectly acceptable for an endpoint to transmit data on the same connection it is using to receive data, so long as

Yon INTERNET-DRAFT - Expires September 2003 5

INTERNET-DRAFT Connection-Oriented Media in SDP March 2003

the other endpoint has advertised its willingness to accept data. Likewise, it is perfectly acceptable for an endpoint to receive data on the same connection it is using to transmit data to the corresponding remote endpoint. In other words, for a bi-directional application-level session, a connection may be used to send data in both directions (contingent to rules outlined in [Section 2.3](#)) as long as one side of the connection is attached to either of the advertised SDP transport addresses.

4.6 Treating UDP and RTP/AVP like Connection Oriented Media

Endpoints MAY specify a direction attribute for UDP or RTP/AVP media. This indicates that the endpoint would like to treat this media as a type of connection-oriented media. (The endpoint may do this to facilitate NAT traversal for example.) Note that for backwards compatibility, an endpoint which can specify direction:active MUST include valid addresses and ports in the SDP as always. If the peer's SDP does not include a direction

attribute, it knows that the peer does not support connection-oriented media, and media exchange will proceed normally, as if connection-oriented media were not offered.

Endpoints that specify `direction:passive` MUST NOT send any media, any packets whatsoever (including control packets such as RTCP), from their passive ports until they receive a packet on these ports and record the source address and port of the sender. The passive endpoint then assumes that the first packet received corresponds to its active peer. From this point onward, passive endpoints MUST send UDP or RTP media from the same port as the port indicated in the `m=` line. Passive endpoints MUST send RTCP media (if any) from the port on which they expect to receive it (typically the RTP port number plus 1).

Endpoints that specify `direction:active` MUST be prepared to receive on the ports from which they send. Once they learn the IP address and port of their peer from the peer's SDP, they SHOULD immediately send some kind of media (even if just comfort noise) to each of these ports. This is so the peer can learn their IP address and port, in order to send media back without additional delay. Effectively, the exchange of the first media packet completes a bi-directional handshake between the active and passive peer.

5 Reconnect Attribute

The preceding description of the `a=direction` attribute has been in the context of using SDP to initiate a session. However, 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, renegotiating the media parameters for a session, etc. 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 reconnect attribute is used to disambiguate these two scenarios, and the syntax is as follows:

`a=reconnect`

SDP containing `a=reconnect` signals that the specified session does NOT refer to an existing connection between the two endpoints. If the endpoints agree to continue the session, the endpoints MUST close the existing connection for the currently negotiated session, and MUST create a new connection according to the `a=direction` attribute in the SDP. If an endpoint receives SDP that contains

a=reconnect, the endpoint's response MUST also contain a=reconnect. Endpoints MUST NOT include a=reconnect in SDP that negotiates the start of a session.

See [section 6](#), "Connection and Listener Lifetime Considerations" for more information on scenarios that are relevant to the a=reconnect attribute.

6 Connection and Listener Lifetime Considerations

6.1 Listener Lifetime

An endpoint that has specified direction:both or direction:passive MUST be ready to accept a connection on the appropriate address and port during the time slot(s) advertised for that session. The endpoint MUST keep the address and port available for incoming connections until either:

- a) The time window for the session has expired, or
- b) The endpoint has received the expected number of incoming connections on that address and port, or
- c) Subsequent exchanges have superceded the SDP that originally advertised the availability of the address and port.

Once the endpoint has determined that a listener is no longer needed on a specific address and port, it SHOULD terminate the listener. The endpoint is then free to re-use the address and port for subsequent session advertisements.

6.2 Connection Lifetime

An endpoint that intends to initiate the connection MUST initiate the connection immediately after it has sufficient information to do so, even if it does not intend to immediately begin sending media to the remote endpoint. This allows media to flow from the remote endpoint.

An endpoint MUST NOT close the connection until the session has expired, been explicitly terminated, or the media stream is redirected to a different address or port.

If the endpoint determines that the connection has been closed, it MAY attempt to re-establish the connection. The decision to do so is application and/or context dependant. If the endpoint opts to

re-establish the connection, it MUST NOT assume that the original address and port advertised by the remote endpoint is still valid. Instead, the endpoint MUST renegotiate the session parameters by exchanging new SDP.

6.3 Session Renegotiation and Connection Lifetime

There are scenarios where SDP is sent by an endpoint in order to renegotiate an existing session. These include muting/unmuting a session, renegotiating the attributes of the media used by the session, or extending the length of a session about to expire. Connection-oriented media introduces some ambiguities into session renegotiation as to when the direction attribute must be obeyed and when it is ignored.

The scenario of extending the duration of an existing session is a good example: in order to extend an existing session, endpoints will typically resend the original SDP with updated time information. In connectionless media the result is no change to the existing media streams. The problem with connection oriented media is that the original SDP will contain a direction attribute which can be construed as a request to create a new connection, as opposed to a request to maintain steady state. To avoid this ambiguity, the following rule SHALL apply to subsequent exchanges of SDP:

If the transport section (the c= and m= lines) combined with the direction attribute of an SDP message describes an existing connection between two endpoints, AND the SDP does not contain a=reconnect, then the endpoints MUST use that connection to carry the media described in the remainder of the message. The endpoints MUST NOT attempt to set up a new connection, regardless of what is specified in the direction attribute.

This disambiguates most session renegotiation scenarios, with the exception of muting. Muting a media stream is accomplished by sending the original session SDP but with an "a=inactive" or "a=sendonly/recvonly" attribute. This is still valid for connection oriented media, with the additional caveat that the endpoints MUST NOT close the connection described by that SDP.

7 Examples

What follows are a number of examples that show the most common usage of the direction attribute combined with TCP-based media descriptions. For the purpose of brevity, the main portion of the session description is omitted in the examples and is assumed to be the following:

```
v=0
o=me 2890844526 2890842807 IN IP4 10.1.1.2
s=Call me using TCP
t=3034423619 3042462419
```

7.1 Example: simple passive/active

An endpoint at 10.1.1.2 signals the availability of a T.38 fax session at port 54111:

```
c=IN IP4 10.1.1.2
m=image 54111 TCP t38
a=direction:passive
```

An endpoint at 10.1.1.1 receiving this description responds with the following:

```
c=IN IP4 10.1.1.1
m=image 9 TCP t38
a=direction:active
```

The endpoint at 10.1.1.1 then initiates the TCP connection to port 54111 at 10.1.1.2.

7.2 Example: simple passive/active with reconnect

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 10.1.1.2 signals this intent with the following SDP:

```
c=IN IP4 10.1.1.2
m=image 54111 TCP t38
a=direction:passive
a=reconnect
```

The a=reconnect attribute informs the endpoint at 10.1.1.1 that this SDP represents the intent to establish a new connection for media transport, rather than continuing with the original connection. Because the endpoint at 10.1.1.1 may not yet be aware that the TCP connection has failed, this eliminates any ambiguity. If 10.1.1.1 agrees to continue the session using a new connection, it responds with:

```
c=IN IP4 10.1.1.1
m=image 9 TCP t38
a=direction:active IN IP4
a=reconnect
```

7.3 Example: agnostic both

An endpoint at 10.1.1.2 signals the availability of a T.38 fax session at TCP port 54111, but is also willing to set up the media stream by initiating the TCP connection:

Yon INTERNET-DRAFT - Expires September 2003 9

INTERNET-DRAFT Connection-Oriented Media in SDP March 2003

```
c=IN IP4 10.1.1.2
m=image 54111 TCP t38
a=direction:both
```

The endpoint at 10.1.1.1 has three choices:

- 1) It can respond with either of the two direction:active descriptions listed in the previous example. In this case the endpoint at 10.1.1.1 must initiate a connection to port 54111 at 10.1.1.2.
- 2) It can respond with a description similar to the following:

```
c=IN IP4 10.1.1.1
m=image 54321 TCP t38
a=direction:passive
```

In this case the endpoint at 10.1.1.2 must initiate a connection to port 54321 at 10.1.1.1.

- 3) It can respond with a description that specifies direction:both, which is covered in the next example.

7.4 Example: redundant both

An endpoint at 10.1.1.2 uses the same description as the previous example:

```
c=IN IP4 10.1.1.2
m=image 54111 TCP t38
a=direction:both
```

Unlike the previous example, the endpoint at 10.1.1.1 responds with the following description:

```
c=IN IP4 10.1.1.1
m=image 54321 TCP t38
a=direction:both
```

This will cause the endpoint at 10.1.1.2 to initiate a connection to

port 54321 at 10.1.1.1, and the endpoint at 10.1.1.1 to initiate a connection to port 54111 at 10.1.1.2. Whichever TCP connection succeeds will be used. If both succeed, one of the connections may be closed as an optimization, using the rules in [section 3.3](#).

In order to minimize the chance that two connections are created, the endpoint at 10.1.1.1 may opt to use the recommendation in [section 3.4](#), which would result in events occurring in the following sequence:

- 1) The endpoint at 10.1.1.2 sends SDP as listed above. The endpoint MUST enable a listener on port 54111 at this time, but is not able to initiate a TCP connection due to the fact

Yon INTERNET-DRAFT - Expires September 2003 10

INTERNET-DRAFT Connection-Oriented Media in SDP March 2003

that it does not have sufficient information from the endpoint at 10.1.1.1.

- 2) The endpoint at 10.1.1.1, upon receiving the SDP, immediately initiates a TCP connection to 10.1.1.2:54111.
- 3) In order to minimize the chance of a duplicate connection, the endpoint at 10.1.1.1 pauses for a short time to allow the endpoint at 10.1.1.2 to receive the TCP connection initiation.
- 4) After the short pause, the endpoint at 10.1.1.1 sends the SDP response as listed above.

The pause in #3 gives the first TCP connection attempt a chance to succeed, since withholding the SDP response deprives the endpoint at 10.1.1.2 of the information it needs to attempt its own TCP connection.

[7.5](#) Example: "Bidirectional" RTP and RTCP

An endpoint at 10.1.1.2 is behind a NAT and does not know its own public address.

```
c=IN IP4 10.1.1.2
m=audio 9 RTP/AVP 0
a=direction:active
```

A peer with a public IP address responds as follows and waits to receive RTP and RTCP packets from its active peer.

```
c=IN IP4 1.2.3.4
m=audio 18240 RTP/AVP 0
a=direction:passive
```

The endpoint at 10.1.1.2 immediately sends RTP from port 9012 to 1.2.3.4 port 18240. A NAT translates the source address to 5.6.7.8 port 1542. The passive endpoint receives this RTP packet and stores this source address. When the passive endpoint wants to send RTP media it sends it back to 5.6.7.8 port 1542. The NAT translates this destination address back to 10.1.1.2 port 9012 and delivers it to the active endpoint.

Likewise the endpoint at 10.1.1.2 immediately sends RTCP from port 9013 to 1.2.3.4:18241. The NAT translates this to 5.6.7.8:1984. The passive endpoint receives the RTCP packet and stores the source address. The passive endpoint sends its RTCP to 5.6.7.8:1984 which is translated back to 10.1.1.2:9013 and delivered to the active endpoint.

8 Security Considerations

See [[SDP](#)] for security and other considerations specific to the Session Description Protocol in general.

| | | |
|----------------|---|------------|
| Yon | INTERNET-DRAFT - Expires September 2003 | 11 |
| INTERNET-DRAFT | Connection-Oriented Media in SDP | March 2003 |

9 IANA Considerations

As recommended by [[SDP](#)] [Appendix B](#), the direction and reconnect attributes described in this document should be registered with IANA, as should the "TCP" and "TLS" protocol identifiers.

Acknowledgements

The author would like to thank Jonathan Rosenberg, Rohan Mahy, Anders Kristensen, Jeorg Ott, Paul Kyzivat, and Robert Fairlie-Cunninghame for their valuable insights and contributions.

Appendix A: Direction Attribute Syntax

This appendix provides an Augmented BNF [[ABNF](#)] grammar for expressing the direction attribute for connection setup. It is intended as an extension to the grammar for the Session Description Protocol, as defined in [[SDP](#)]. Specifically, it describes the syntax for the new "connection-setup" attribute field, which MAY be either a session-level or media-level attribute.

```
connection-setup = "direction" ":" direction-spec
```

```
direction-spec = "both" / "active" / "passive"
```

```
reconnect-attribute = "reconnect"
```

References

- [ABNF] D. Crocker, P. Overell, "Augmented BNF for Syntax Specifications: ABNF," [RFC 2234](#), November 1997
- [SDP] M. Handley, V. Jacobson, "SDP: Session Description Protocol," [RFC 2327](#), April 1998

- [T38] International Telecommunication Union, "Procedures for Real-Time Group 3 Facsimile Communications over IP Networks," Recommendation T.38, June 1998
- [TLS] T. Dierks, C. Allen, "The TLS Protocol," [RFC 2246](#), January 1999
- [UTF-8] F. Yergeau, "UTF-8, a transformation format of Unicode and ISO 10646," [RFC 2044](#), October 1996

Author's Address

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

Phone: (603) 324-4100
EMail: yon@dialout.net

Full Copyright Statement

Copyright (C) The Internet Society (2003). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this

Yon INTERNET-DRAFT - Expires September 2003 13

INTERNET-DRAFT Connection-Oriented Media in SDP March 2003

document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS 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."