



# XMPP

## XEP-0343: Signaling WebRTC datachannels in Jingle

Jens Bavendiek

<mailto:bavendiek@dbis.rwth-aachen.de>

2014-07-15

Version 0.2

Status	Type	Short Name
Experimental	Standards Track	jingle-webrtc-datachannels

This specification defines how to use the ICE-UDP Jingle transport method to send media data using WebRTC DataChannels, so technically uses DTLS/SCTP on top of the Interactive Connectivity Establishment (ICE) methodology, which provides robust NAT traversal for media traffic.

# Legal

## Copyright

This XMPP Extension Protocol is copyright © 1999 – 2017 by the [XMPP Standards Foundation](#) (XSF).

## Permissions

Permission is hereby granted, free of charge, to any person obtaining a copy of this specification (the "Specification"), to make use of the Specification without restriction, including without limitation the rights to implement the Specification in a software program, deploy the Specification in a network service, and copy, modify, merge, publish, translate, distribute, sublicense, or sell copies of the Specification, and to permit persons to whom the Specification is furnished to do so, subject to the condition that the foregoing copyright notice and this permission notice shall be included in all copies or substantial portions of the Specification. Unless separate permission is granted, modified works that are redistributed shall not contain misleading information regarding the authors, title, number, or publisher of the Specification, and shall not claim endorsement of the modified works by the authors, any organization or project to which the authors belong, or the XMPP Standards Foundation.

## Warranty

## NOTE WELL: This Specification is provided on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. ##

## Liability

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall the XMPP Standards Foundation or any author of this Specification be liable for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising from, out of, or in connection with the Specification or the implementation, deployment, or other use of the Specification (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if the XMPP Standards Foundation or such author has been advised of the possibility of such damages.

## Conformance

This XMPP Extension Protocol has been contributed in full conformance with the XSF's Intellectual Property Rights Policy (a copy of which can be found at <https://xmpp.org/about/xsf/ipr-policy>) or obtained by writing to XMPP Standards Foundation, P.O. Box 787, Parker, CO 80134 USA).

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Use Cases</b>	<b>1</b>
<b>3</b>	<b>Jingle Conformance</b>	<b>1</b>
<b>4</b>	<b>Protocol Description</b>	<b>2</b>
4.1	Session Initiation . . . . .	2
4.2	Syntax . . . . .	4
4.3	Response . . . . .	4
4.4	Adding a new channel . . . . .	6
<b>5</b>	<b>Security Considerations</b>	<b>7</b>
<b>6</b>	<b>Acknowledgements</b>	<b>7</b>
<b>7</b>	<b>IANA Considerations</b>	<b>7</b>
<b>8</b>	<b>XMPP Registrar Considerations</b>	<b>7</b>
8.1	Protocol Namespaces . . . . .	7
8.2	Protocol Versioning . . . . .	8
8.3	SCTP Mapping . . . . .	8
<b>9</b>	<b>XML Schema</b>	<b>8</b>

## 1 Introduction

This document specifies a method for establishing a direct connection using SCTP channels over DTLS. As WebRTC supports the SCTP partial reliability extension (see [RFC 3758](#)<sup>1</sup>), the connection established can either be reliable or partial-reliable. However due to the wider use (e.g. [Jingle File Transfer \(XEP-0234\)](#)<sup>2</sup>) and easier maintenance, this transport method defines the mandatory use of reliable and in-order-delivery datachannels (thus a streaming transport). If the need for partial-reliability is there, a second specification for datagram type datachannels might be created. The establishment of this connection is managed by [Jingle ICE-UDP Transport Method \(XEP-0176\)](#)<sup>3</sup>. The establishment of security provided by DTLS is described in [Use of DTLS-SRTP in Jingle Sessions \(XEP-0320\)](#)<sup>4</sup>.

## 2 Use Cases

There are several use cases. Given below are a few examples:

1. Two parties engaged in a WebRTC-based video conference want to share a file via [Jingle File Transfer \(XEP-0234\)](#)<sup>5</sup>
2. A browser application that uses XMPP for communication wants to support file-transfer
3. A browser based application wants to send a file to a non-browser client

## 3 Jingle Conformance

In accordance with Section 10 of XEP-0166, this document specifies the following information related to the Jingle DTLS/SCTP transport method:

1. The transport negotiation process is defined in the [Protocol Description](#) section of this document.
2. The semantics of the <sctpmap/> element are defined in the [Syntax](#) section of this document.

---

<sup>1</sup>RFC 3758: Stream Control Transmission Protocol (SCTP) Partial Reliability Extension <<http://tools.ietf.org/html/rfc3758>>.

<sup>2</sup>XEP-0234: Jingle File Transfer <<https://xmpp.org/extensions/xep-0234.html>>.

<sup>3</sup>XEP-0176: Jingle ICE-UDP Transport Method <<https://xmpp.org/extensions/xep-0176.html>>.

<sup>4</sup>XEP-0320: Use of DTLS-SRTP in Jingle Sessions <<https://xmpp.org/extensions/xep-0320.html>>.

<sup>5</sup>XEP-0234: Jingle File Transfer <<https://xmpp.org/extensions/xep-0234.html>>.

## 4 Protocol Description

This protocol requires the Stream Control Transmission Protocol (SCTP) to run within the security context of the Datagram Transport Layer Security (DTLS) protocol. As determined by RTCWeb Data Channels running SCTP on top of DTLS is preferred, as in this order the control messages are encrypted as well and the DTLS channel can be shared with several applications

### 4.1 Session Initiation

In order for the initiator in a Jingle exchange to start the negotiation, it sends a Jingle "session-initiate" stanza that includes at least one content type, as described in XEP-0166. If the initiator wishes to negotiate the SCTP transport method for an application format, it MUST include a <sctpmap/> child element qualified by the 'urn:xmpp:jingle:transports:dtls-sctp:1' namespace (see [Namespace Versioning](#) regarding the possibility of incrementing the version number). The <transport/> element SHOULD in turn contain one <candidate/> element for each of the initiator's higher-priority transport candidates as determined in accordance with the ICE methodology, but MAY instead be empty (with each candidate to be sent as the payload of a transport-info message).

Listing 1: Initiation

```
<iq from='romeo@montague.lit/orchard'
  id='ixt174g9'
  to='juliet@capulet.lit/balcony'
  type='set'>
  <jingle xmlns='urn:xmpp:jingle:1'
    action='session-initiate'
    initiator='romeo@montague.lit/orchard'
    sid='a73sjvkla37jfea'>
    <content creator='initiator' name='file-552
      da749930852c69ae5d2141d3766b1'>
      <description xmlns='urn:xmpp:jingle:apps:file-transfer:3'>
        <offer>
          <file>
            <date>1969-07-21T02:56:15Z</date>
            <desc>This is a test. If this were a real file...</desc>
            <name>test.txt</name>
            <range/>
            <size>1022</size>
            <hash xmlns='urn:xmpp:hashes:1' algo='sha-1'>552
              da749930852c69ae5d2141d3766b1</hash>
          </file>
        </offer>
      </description>
      <transport xmlns='urn:xmpp:jingle:transports:ice-udp:1'
        pwd='asd88fgpdd777uzjYhagZg'
```

```

        ufrag='8hhy'>
<sctpmap xmlns='urn:xmpp:jingle:transports:dtls-sctp:1' number
  ='5000' protocol='webrtc-datachannel' streams='1024' />
<fingerprint mlns='urn:xmpp:jingle:apps:dtls:0' hash='sha-256'
  setup='actpass'>
  02
    :1A:CC:54:27:AB:EB:9C:53:3F:3E:4B:65:2E:7D:46:3F:54:42:CD:54:F1:7A:03:

</fingerprint>
<candidate component='1'
  foundation='1'
  generation='0'
  id='el0747fg11'
  ip='10.0.1.1'
  network='1'
  port='8998'
  priority='2130706431'
  protocol='udp'
  type='host' />
<candidate component='1'
  foundation='2'
  generation='0'
  id='y3s2b30v3r'
  ip='192.0.2.3'
  network='1'
  port='45664'
  priority='1694498815'
  protocol='udp'
  rel-addr='10.0.1.1'
  rel-port='8998'
  type='srflx' />
</transport>
</content>
</jingle>
</iq>

```

As seen in the example, there is no notion about DataChannel internals. The only things negotiated here are the port on which the SCTP Stream runs and the name of the DataChannel. Further negotiation is happening through the [WebRTC Data Channel Protocol](#)<sup>6</sup>, thus the DataChannel is identified through the content name.

For some uses like negotiated datachannels, it might be required to explicitly signal the channel properties. This is done by including a <channel/> element qualified by the 'urn:xmpp:jingle:transports:webrtc-datachannel:0' namespace (see [Namespace Versioning](#) regarding the possibility of incrementing the version number) as child of the <transport/> for each channel that has to be signaled. An abbreviated example follows:

<sup>6</sup>WebRTC Data Channel Protocol <<http://tools.ietf.org/html/draft-jesup-rtcweb-data-protocol-04>>.

Listing 2: Initiation (with explicit channel signaling)

```
[...]
<transport xmlns='urn:xmpp:jingle:transports:ice-udp:1'
  pwd='asd88fgpdd777uzjYhagZg'
  ufrag='8hhy'>
  <sctpmap xmlns='urn:xmpp:jingle:transports:dtls-sctp:1' number
    ='5000' protocol='webrtc-datachannel' streams='1024' />
  <channel xmlns='urn:xmpp:jingle:transports:webrtc-
    datachannel:0' id="1" negotiated="true"/>
  <channel xmlns='urn:xmpp:jingle:transports:webrtc-
    datachannel:0' id="3" negotiated="true"/>
  <fingerprint xmlns='urn:xmpp:jingle:apps:dtls:0' hash='sha-256'
    setup='actpass'>
[...]
```

## 4.2 Syntax

The values and names for the <sctpmap/> element are taken from the corresponding SDP. The attributes of the <sctpmap/> element are as follows.

Name	Description	SDP Syntax	Example
port	A port number where the channel is to be contacted	Port value in a=sctpmap line	5000
protocol	Media format	Protocol value in a=sctpmap line	webrtc-datachannel
streams	Number of supported incoming streams	Stream value in a=sctpmap line	16

The attributes of the <channel/> element map directly to the attributes of the [RTCDataChannelInit](#) dictionary as defined in [WebRTC 1.0: Real-time Communication Between Browsers](#)<sup>7</sup>.

## 4.3 Response

As described in XEP-0166, to acknowledge receipt of the session initiation request, the responder immediately returns an IQ-result.

Listing 3: Responder acknowledges receipt of session-initiate request

<sup>7</sup>WebRTC 1.0: Real-time Communication Between Browsers <<http://www.w3.org/TR/2013/WD-webrtc-20130910/>>.

```
<iq from='juliet@capulet.lit/balcony'
  id='ixt174g9'
  to='romeo@montague.lit/orchard'
  type='result'/>
```

Depending on the application type, a user agent controlled by a human user might need to wait for the user to affirm a desire to proceed with the session before continuing. When the user agent has received such affirmation (or if the user agent can automatically proceed for any reason, e.g. because no human intervention is expected or because a human user has configured the user agent to automatically accept sessions with a given entity), it returns a Jingle session-accept message. This message MUST contain a <transport/> element qualified by the 'urn:xmpp:jingle:transports:ice-udp:1' namespace, which SHOULD in turn contain one <sctpmap/> element qualified by the 'urn:xmpp:jingle:transports:dtls-sctp:1' namespace for the channel used by the responder.

Note: See the [Security Considerations](#) section of this document regarding the exposure of IP addresses by the responder's client.

Listing 4: Responder accepts the session request

```
<iq from='juliet@capulet.lit/balcony'
  id='rw782g55'
  to='romeo@montague.lit/orchard'
  type='set'>
  <jingle xmlns='urn:xmpp:jingle:1'
    action='session-accept'
    initiator='romeo@montague.lit/orchard'
    responder='juliet@capulet.lit/balcony'
    sid='a73sjvkl37jfea'>
    <content creator='initiator' name='file-552
      da749930852c69ae5d2141d3766b1'>
      <description xmlns='urn:xmpp:jingle:apps:file-transfer:3'>
        <offer>
          <file>
            <date>1969-07-21T02:56:15Z</date>
            <desc>This is a test. If this were a real file...</desc>
            <name>test.txt</name>
            <range/>
            <size>1022</size>
            <hash xmlns='urn:xmpp:hashes:1' algo='sha-1'>552
              da749930852c69ae5d2141d3766b1</hash>
          </file>
        </offer>
      </description>
      <transport xmlns='urn:xmpp:jingle:transports:ice-udp:1'
        pwd='YH75Fviy6338Vbrhr1p8Yh'
        ufrag='9uB6'>
      <fingerprint xmlns='urn:xmpp:jingle:apps:dtls:0' hash='sha-256'
        setup='actpass'>
```



```

02
   :1A:CC:54:27:AB:EB:9C:53:3F:3E:4B:65:2E:7D:46:3F:54:42:CD:54:F1:7A:03:

</fingerprint>
<sctpmap xmlns='urn:xmpp:jingle:transports:dtls-sctp:1' number
  = '5000' protocol='webrtc-datachannel' streams='1024' />
<candidate component='1'
  foundation='1'
  generation='0'
  id='or2ii2syr1'
  ip='192.0.2.1'
  network='0'
  port='3478'
  priority='2130706431'
  protocol='udp'
  type='host' />
</transport>
</content>
</jingle>
</iq>

```

The initiator acknowledges the Jingle session-accept.

Listing 5: Initiator acknowledges session-accept.

```

<iq from='juliet@capulet.lit/balcony'
  id='ixt174g9'
  to='romeo@montague.lit/orchard'
  type='result' />

```

#### 4.4 Adding a new channel

Even after media has begun to flow, either party MAY continue to send additional channels to the other party (e.g., because the user wants to transmit a new file). Such events are shared by sending a content-add message.

Listing 6: Initiator sends a content-add

```

<iq from='romeo@montague.lit/orchard'
  id='uh3g1f48'
  to='juliet@capulet.lit/balcony'
  type='set'>
  <jingle xmlns='urn:xmpp:jingle:1'
    action='content-add'
    initiator='romeo@montague.lit/orchard'
    sid='a73sjvkla37jfea'>
    <content creator='initiator' name='this-is-the-new-content'>

```

```
<description xmlns='urn:xmpp:jingle:apps:some-file' media='file'
  />
<transport xmlns='urn:xmpp:jingle:transports:ice-udp:1' />
</content>
</jingle>
</iq>
```

The receiving party MUST acknowledge receipt.

Listing 7: Recipient acknowledges receipt

```
<iq from='juliet@capulet.lit/balcony'
  id='uh3g1f48'
  to='romeo@montague.lit/orchard'
  type='result' />
```

## 5 Security Considerations

By design is the sent media already encrypted by the DTLS layer

## 6 Acknowledgements

Special thanks to István Koren at i5 and their [XMPP Experience](#)

Thanks to Philipp Hancke and Tobias Markmann for their help and support and all the XEP-Authors for their examples.

## 7 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)<sup>8</sup>.

## 8 XMPP Registrar Considerations

### 8.1 Protocol Namespaces

This specification defines the following XML namespace:

---

<sup>8</sup>The Internet Assigned Numbers Authority (IANA) is the central coordinator for the assignment of unique parameter values for Internet protocols, such as port numbers and URI schemes. For further information, see <http://www.iana.org/>.

- urn:xmpp:jingle:transports:dtls-sctp:1
- urn:xmpp:jingle:transports:webrtc-datachannel:0

The XMPP Registrar <sup>9</sup> includes the foregoing namespace in its registry at <https://xmpp.org/registrar/namespaces.html>, as governed by XMPP Registrar Function (XEP-0053) <sup>10</sup>.

## 8.2 Protocol Versioning

If the protocol defined in this specification undergoes a revision that is not fully backwards-compatible with an older version, the XMPP Registrar shall increment the protocol version number found at the end of the XML namespaces defined herein, as described in Section 4 of XEP-0053.

## 8.3 SCTP Mapping

The registry submission is as follows.

```
<var>
  <name>IETF Draft MMUSIC</name>
  <desc>
    Signals support for SCTP in the SDP as described in draft-ietf-
      mmusic-sctp-sdp-05
  </desc>
  <doc>XEP-SCTP</doc>
</var>
```

## 9 XML Schema

```
<?xml version='1.0' encoding='UTF-8'?>
<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='urn:xmpp:jingle:transports:dtls-sctp:1'
  xmlns='urn:xmpp:jingle:transports:dtls-sctp:1'
  elementFormDefault='qualified'>
  <xs:annotation>
```

<sup>9</sup>The XMPP Registrar maintains a list of reserved protocol namespaces as well as registries of parameters used in the context of XMPP extension protocols approved by the XMPP Standards Foundation. For further information, see <https://xmpp.org/registrar/>.

<sup>10</sup>XEP-0053: XMPP Registrar Function <https://xmpp.org/extensions/xep-0053.html>.

```

    <xs:documentation>
      The protocol documented by this schema is defined in
      XEP-0343: http://www.xmpp.org/extensions/xep-0343.html
    </xs:documentation>
  </xs:annotation>

  <xs:element name='sctpmap'>
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base='xs:string'>
          <xs:attribute name='number' type='xs:unsignedByte' use='
            required' />
          <xs:attribute name='protocol' type='xs:string' use='required
            '/>
          <xs:attribute name='streams' type='xs:positiveInteger' use='
            optional' />
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
</xs:schema>

<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='urn:xmpp:jingle:transports:webrtc-datachannel:0'
  xmlns='urn:xmpp:jingle:transports:webrtc-datachannel:0'
  elementFormDefault='qualified'>

  <xs:annotation>
    <xs:documentation>
      The protocol documented by this schema is defined in
      XEP-0343: http://www.xmpp.org/extensions/xep-0343.html
    </xs:documentation>
  </xs:annotation>

  <xs:element name='channel'>
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base='xs:string'>
          <xs:attribute name='id' type='xs:unsignedShort' use='
            optional' />
          <xs:attribute name='maxPacketLifeTime' type='
            xs:unsignedShort' use='optional' />
          <xs:attribute name='maxRetransmits' type='xs:unsignedShort'
            use='optional' />
          <xs:attribute name='negotiated' type='xs:boolean' use='
            optional' default='false' />
          <xs:attribute name='ordered' type='xs:boolean' use='optional
            ' default='true' />
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

```
        <xs:attribute name='protocol' type='xs:string' use='optional' />
    </xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
</xs:schema>
```