MMUSIC
Internet-Draft
Intended status: Standards Track
Expires: September 6, 2011

A. Begen
Y. Cai
H. Ou
Cisco

March 5, 2011

Redundancy Grouping Semantics in the Session Description Protocol
draft-begen-mmusic-redundancy-grouping-00

Abstract

Packet loss is undesirable for real-time multimedia sessions, but it is not avoidable due to congestion or other unplanned network outages. This is especially the case for IP multicast networks. One technique to recover from packet loss without incurring unbounded delay for all the receivers is to send redundant streams. This document defines the semantics for grouping RTP-encapsulated redundant streams in the Session Description Protocol (SDP). The semantics defined in this document are to be used with the SDP Grouping Framework [RFC5888]. SSRC-level (Synchronization Source) grouping semantics are also defined in this document for RTP streams using SSRC multiplexing.

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 September 6, 2011.

Copyright Notice

Table of Contents

1. Introduction

RTP [RFC3550]transport is widely used today for delivering real-time multimedia streams. Most of the applications also rely on IP multicast to reach many receivers efficiently.

While the combination proves successful, there does exist a weakness. As [RFC2354] noted, packet loss is not avoidable. This might be due to congestion, it might also be a result of an unplanned outage caused by a flapping link, link or interface failure, a software bug, or a maintenance person accidentally cutting the wrong fiber. Since UDP does not provide any means for detecting loss and retransmitting packets, it leaves up to the RTP or the applications to detect and recover from the loss. For retransmission-based recovery, one example is described by [RFC4588].

In this document, we describe a technique that involves transmitting redundant RTP-encapsulated streams to overcome packet loss. Variations of the technique have already been implemented and deployed today [IC2011]. We also describe the semantics needed in the Session Description Protocol (SDP) [RFC4566] to support this technique.

A work-in-progress draft specification [I-D.singh-avtcore-mprtp] proposes changes to the RTP protocol so that a single RTP session can benefit from using multiple paths between two endpoints (to increase the aggregated throughput and improve reliability). While we also discuss spatial diversity in our specification, we use diverse paths solely for sending redundant streams. For our purposes, we do not require changes in the RTP protocol but if any changes or additions will be required for RTP Control Protocol (RTCP), they will be documented in the future or separately.

2. Requirements Notation

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 [RFC2119].

3. Dual Streaming

Dual streaming refers to a technique that involves transmitting two RTP streams of the same content, with each stream itself capable of supporting the playback when there is no packet loss. Therefore, adding an additional stream provides a protection against packet

loss. The level of protection depends on how the packets are built and sent.

It is important to note that the technique and specification described by this document can easily be extended to support cases when more than two streams are desired. But triple, quadruple, or more, streaming is rarely used in practice, and therefore the solutions developed by this document are not optimized for them.

3.1. (Routing-Plane) Identical Streams

From a routing perspective, two streams are considered identical if their following two fields are the same since they will be both routed over the same path:

- IP Source Address
- IP Destination Address

Two RTP streams can be routing-plane identical, however, they might differ in their UDP destination ports, Synchronization Sources (SSRC) or payload types. For example, two such RTP streams also sharing the same UDP destination port but differing in their SSRCs can carry the same payload in their respective RTP packets with common sequence numbers. This allows the receiver (or any other node responsible for duplicate suppression) to identify the duplicate packets. Alternatively, each RTP packet can also be duplicated and the receiver identifies them based on the sequence numbers.

In these cases (where the two RTP streams are routing-plane identical and share the same UDP destination port), there will be only one "m" line in the SDP description regardless of how many redundant streams are generated. Thus, the SDP Grouping Framework [RFC5888] cannot be used. Instead, either the ’ssrc-group’ attribute [RFC5576] or the ’temporal-interleaving’ attribute [I-D.begen-mmusic-temporal-interleaving] has to be used to describe the redundancy relations.

If the two routing-plane identical RTP streams have different UDP destination ports, there will be two "m" lines in the SDP description and in this case, new grouping semantics are needed to describe the redundancy relations. Based on this grouping, the receiver or the node responsible for duplicate suppression can look into various RTP related fields to identify and suppress the duplicate packets.

3.2. Single RTP Session

If the two streams use different IP source addresses, but the same IP destination address and UDP port, the two streams are considered to be using the same RTP session. These streams can be routed over diverse or identical paths. Similar to the above, the receiver or the node responsible for duplicate suppression can look into various RTP related fields to identify and suppress the duplicate packets.

Note that in source-specific multicast (SSM) scenarios, the host has to join the two streams separately via Internet Group Management Protocol (IGMP) version 3 [RFC3376] or the Multicast Listener Discovery Protocol (MLD) version 2 [RFC3810] since the source addresses are different.

3.3. Multiple RTP Sessions

When dual streaming is done via multiple RTP sessions, the two streams use different IP destination addresses and/or UDP destination ports. In this case, new grouping semantics are needed to describe the redundancy relations.

3.4. Summary

Having described the characteristics of the streams, one can reach the following conclusions:

1. When two routing-plane identical streams are used, the two streams will have identical IP headers. This makes it impractical to forward the packets onto different paths. In order to minimize packet loss, the packets belonging to one stream are often interleaved with packets belonging to the other, and with a delay, so that if there is a packet loss, such a delay would allow the same packet from the other stream to reach the receivers because the chances that the same packet is lost in transit again is often small. This is what is also known as Temporal Interleaving or Temporal Redundancy [I-D.begen-mmusic-temporal-interleaving] [IC2011]. This approach can be used with all three types of dual streaming described in Section 3.1, Section 3.2 and Section 3.3.

2. If the two streams have different IP headers, an additional opportunity arises in that one is able to build a network, with physically diverse paths, to deliver the two streams concurrently to the intended receivers. This reduces the delay when packet loss occurs and needs to be recovered. Additionally, it also further reduces chances for packet loss. An unrecoverable loss happens only when two network failures happen in such a way that

the same packet is affected on both paths. This is referred to as Spatial Diversity or Spatial Redundancy [IC2011]. The techniques used to build diverse paths are beyond the scope of this document.

Note that spatial redundancy often offers less delay in recovering from packet loss provided that the forwarding delay of the network paths are more or less the same. For both temporal and spatial redundancy approaches, packet misordering might still happen and needs to be handled using the RTP sequence numbers.

To summarize, dual streaming allows an application and a network to work together to provide a near zero-loss transport with a bounded or minimum delay. The additional advantage includes a predictable bandwidth overhead that is proportional to the minimum bandwidth needed for the multimedia session, but independent of the number of receivers experiencing a packet loss and requesting a retransmission. For a survey and comparison of similar approaches, refer to [IC2011].

4. Redundancy Grouping

4.1. "RED" Grouping Semantics

Each "a=group" line is used to indicate an association relationship between the redundant streams. The streams included in one "a=group" line are called a Redundancy Group.

Using the framework in [RFC5888], this document defines "RED" as the grouping semantics for redundant streams.

The "a=group:RED" semantics MUST be used to group the redundant streams except when the streams are specified in the same media description, i.e., in the same "m" line (See Section 4.2) or when the 'interleaving-period' attribute is used as defined in [I-D.begen-mmusic-temporal-interleaving].

4.2. RED Grouping for SSRC-Multiplexed RTP Streams

[RFC5576] defines an SDP media-level attribute, called 'ssrc-group', for grouping the RTP streams that are SSRC multiplexed and carried in the same RTP session. The grouping is based on the SSRC identifiers. Since SSRC-multiplexed RTP streams are defined in the same "m" line, the 'group' attribute cannot be used.

This section specifies how redundancy is used with SSRC-multiplexed streams using the 'ssrc-group' attribute [RFC5576].

The semantics of "RED" for the 'ssrc-group' attribute are the same as the one defined for the 'group' attribute except that the SSRC identifiers are used to designate the redundancy grouping associations: a=ssrc-group:RED *(SP ssrc-id) [RFC5576].

The SSRC identifiers for the RTP streams that are carried in the same RTP session MUST be unique per [RFC3550]. However, the SSRC identifiers are not guaranteed to be unique among different RTP sessions. Thus, the 'ssrc-group' attribute MUST only be used at the media level [RFC5576].

4.3. SDP Offer/Answer Model Considerations

When offering redundancy grouping using SDP in an Offer/Answer model [RFC3264], the following considerations apply.

A node that is receiving an offer from a sender may or may not understand line grouping. It is also possible that the node understands line grouping but it does not understand the "RED" semantics. From the viewpoint of the sender of the offer, these cases are indistinguishable.

When a node is offered a session with the "RED" grouping semantics but it does not support line grouping or the redundancy grouping semantics, as per [RFC5888], the node responds to the offer either (1) with an answer that ignores the grouping attribute or (2) with a refusal to the request (e.g., 488 Not Acceptable Here or 606 Not Acceptable in SIP).

In the first case, the original sender of the offer must send a new offer without any redundancy grouping. In the second case, if the sender of the offer still wishes to establish the session, it should retry the request with an offer without the redundancy grouping. This behavior is specified in [RFC5888].

5. SDP Examples

5.1. Single RTP Session with SSM

In this example, the redundant streams use the same IP destination address (232.252.0.1) but they are sourced from different addresses (198.51.100.1 and 198.51.100.2). Thus, the host needs to join both SSM sessions separately. The important requirement here is that the RTP packets with the same sequence numbers in each RTP stream carries the same payload and after duplicate suppression, a single RTP stream is delivered to the application.

```
v=0
o=ali 1122334455 1122334466 IN IP4 red.example.com
s=RED Grouping Semantics
t=0 0
m=video 30000 RTP/AVP 100 101
c=IN IP4 232.252.0.1/127
a=source-filter:incl IN IP4 232.252.0.1 198.51.100.1 198.51.100.2
a=rtpmap:100 MP2T/90000
a=rtpmap:101 MP2T/90000
a=ssrc:1000 cname:ch1@example.com
a=ssrc:2000 cname:ch1@example.com
a=ssrc-group:RED 1000 2000
a=mid:Group1
```

Note that in actual use, SSRC values, which are random 32-bit numbers, can be much larger than the ones shown in this example. Also, note that before receiving an RTP packet for each stream, the receiver cannot know which SSRC identifier is associated with which payload type.

5.2. Multiple RTP Sessions with Different Destination Addresses

In this example, the redundant streams have different IP destination addresses. The example shows the same UDP port number and IP source addresses, but either or both can be different for the two streams.

```
v=0
o=ali 1122334455 1122334466 IN IP4 red.example.com
s=RED Grouping Semantics
t=0 0
a=group:RED S1 S2
m=video 30000 RTP/AVP 100
c=IN IP4 233.252.0.1/127
a=source-filter:incl IN IP4 233.252.0.1 198.51.100.1
a=rtpmap:100 MP2T/90000
a=mid:S1
m=video 30000 RTP/AVP 101
c=IN IP4 233.252.0.2/127
a=source-filter:incl IN IP4 233.252.0.2 198.51.100.1
a=rtpmap:101 MP2T/90000
a=mid:S2
```

5.3. Multiple RTP Sessions with the Same Destination Address

In this example, the redundant streams have the same IP source and destination addresses but different UDP port numbers.

```
v=0
o=ali 1122334455 1122334466 IN IP4 red.example.com
s=RED Grouping Semantics
t=0 0
a=group:RED S1 S2
m=video 30000 RTP/AVP 100
c=IN IP4 233.252.0.1/127
a=source-filter:incl IN IP4 233.252.0.1 198.51.100.1
a=rtpmap:100 MP2T/90000
a=mid:S1
m=video 40000 RTP/AVP 101
c=IN IP4 233.252.0.1/127
a=source-filter:incl IN IP4 233.252.0.1 198.51.100.1
a=rtpmap:101 MP2T/90000
a=mid:S2
```

6. Performance Evaluation and Reporting

Editor's note: TBD

7. Security Considerations

There is a weak threat for the receiver that the redundancy grouping can be modified to indicate relationships that do not exist. Such attacks might result in failure of the redundancy mechanisms, and/or mishandling of the media streams by the receivers.

In order to avoid attacks of this sort, the SDP description needs to be integrity protected and provided with source authentication. This can, for example, be achieved on an end-to-end basis using S/MIME [RFC5652] [RFC5751] when the SDP is used in a signaling packet using MIME types (application/sdp). Alternatively, HTTPS [RFC2818] or the authentication method in the Session Announcement Protocol (SAP) [RFC2974] could be used as well.

8. IANA Considerations

This document registers the following semantics with IANA in Semantics for the 'group' SDP Attribute under SDP Parameters:

Note to the RFC Editor: In the following registrations, please replace "XXXX" with the number of this document prior to publication as an RFC.

Semantics	Token	Reference
Redundancy	RED	[RFCXXXX]

This document also registers the following semantics with IANA in Semantics for the 'ssrc-group' SDP Attribute under SDP Parameters:

Token	Semantics	Reference
RED	Redundancy	[RFCXXXX]

9. Acknowledgments

The authors would like to thank Bill Ver Steeg, Dave Oran and Toerless Eckert for their inputs and suggestions.

10. References

10.1. Normative References

[I-D.begen-mmusic-temporal-interleaving] Begen, A., Cai, Y., and H. Ou, "Temporal Interleaving Attribute in the Session Description Protocol", draft-begen-mmusic-temporal-interleaving-00 (work in progress), March 2011.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

[RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session

Description Protocol", RFC 4566, July 2006.

[RFC5576] Lennox, J., Ott, J., and T. Schierl, "Source-Specific Media Attributes in the Session Description Protocol (SDP)", RFC 5576, June 2009.

[RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework", RFC 5888, June 2010.

10.2. Informative References

[I-D.singh-avtcore-mprtp] Singh, V., Karkkainen, T., Ott, J., Ahsan, S., and L. Eggert, "Multipath RTP (MPRTP)", draft-singh-avtcore-mprtp-00 (work in progress), February 2011.

[IC2011] Evans, J., Begen, A., Greengrass, J., and C. Filsfils, "Towards Lossless Video Transport (in submission to IEEE Internet Computing)", 2011.

[RFC2354] Perkins, C. and O. Hodson, "Options for Repair of Streaming Media", RFC 2354, June 1998.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[RFC2974] Handley, M., Perkins, C., and E. Whelan, "Session Announcement Protocol", RFC 2974, October 2000.

[RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R. Hakenberg, "RTP Retransmission Payload Format", RFC 4588, July 2006.

[RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, September 2009.

[RFC5751] Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", RFC 5751, January 2010.

Authors' Addresses

Ali Begen
Cisco
181 Bay Street
Toronto, ON M5J 2T3
Canada

Email: abegen@cisco.com

Yiqun Cai
Cisco
170 W. Tasman Dr.
San Jose, CA 95134
USA

Email: ycai@cisco.com

Heidi Ou
Cisco
170 W. Tasman Dr.
San Jose, CA 95134
USA

Email: hou@cisco.com

MMUSIC
Internet-Draft
Intended status: Standards Track
Expires: September 13, 2011

A. Begen
Y. Cai
H. Ou
Cisco

March 12, 2011

Temporal Interleaving Attribute in the Session Description Protocol
draft-begen-mmusic-temporal-interleaving-01

Abstract

A straightforward approach to provide protection against network outages (or packet losses) with a longest duration of T time units is to simply duplicate the original packets and send each copy separated in time by at least T time units. This approach is commonly referred to as Temporal Redundancy or Temporal Interleaving. This document defines an attribute to indicate the presence of temporally redundant media streams and the interleaving period in the Session Description Protocol.

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 September 13, 2011.

Copyright Notice

Table of Contents

1. Introduction

Consider that a media sender transmits an original source packet and transmits its duplicate after an interleaving period following the original transmission. If a network outage hits the original transmission, the expectation is that the second transmission arrives at the receiver. Alternatively, the second transmission may be hit by an outage or gets dropped, and the original transmission completes successfully. On the receiver side, both transmissions can also arrive and in that case, the receiver (or the node that does the duplicate suppression) needs to identify the duplicate packet(s) and discard them appropriately, producing a duplication-free stream.

Temporal interleaving can be used in a variety of multimedia applications where there is sufficient bandwidth for the duplicated traffic and the application can tolerate the delay caused by interleaving. One particular use case is to improve the reliability of real-time video feeds inside a core IP network [IC2011]. Compared to other popular redundancy approaches such as Forward Error Correction (FEC) [I-D.ietf-fecframe-framework] and redundant data encoding (e.g., [RFC2198]), temporal interleaving is quite easy to implement since it does not require any special type of encoding or decoding.

For duplicate suppression, the receiver has to be able to identify the identical packets. This is straightforward for media packets that carry one or more unique identifiers such as the sequence number field in RTP header [RFC3550]. The receiver can also use alternative approaches to compare the incoming packets and discard the duplicate ones.

In this specification, we are not concerned about how the sender should determine the interleaving period or how the receiver can suppress the duplicate packets. Rather, we introduce a new attribute for the Session Description Protocol (SDP) [RFC4566] that indicates that the media stream is to be sent two or more times using the interleaving approach and also indicates the interleaving period for each additional duplication.

In practice, more than two redundant streams for temporal interleaving are unlikely to be used since the additional delay and increased overhead are not easily justified. However, we define the new attribute in a general way so that it could be used with more than two redundant streams if needed. The new attribute is applicable to both RTP and non-RTP streams.

2. Requirements Notation

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 [RFC2119].

3. The 'interleaving-period' Attribute

The following ABNF [RFC5234] syntax formally describes the 'interleaving-period' attribute:

```
interleaving-attribute = "a=interleaving-period:" periods CRLF
periods                = period *( ":" period)
period                 = 1*DIGIT ; in milliseconds
```

Figure 1: ABNF syntax for the 'interleaving-period' attribute

The 'interleaving-period' attribute is defined as both a media-level and session-level attribute. It specifies the interleaving duration in milliseconds (ms).

4. SDP Examples

In the example below, the multicast stream is duplicated with an interleaving period of 100 ms.

```
v=0
o=ali 1122334455 1122334466 IN IP4 red.example.com
s=Temporal Interleaving
t=0 0
m=video 30000 RTP/AVP 100
c=IN IP4 233.252.0.1/127
a=source-filter:incl IN IP4 233.252.0.1 198.51.100.1
a=rtpmap:100 MP2T/90000
a=interleaving-period:100
a=mid:S1
```

In the second example below, the multicast stream is duplicated twice. 50 ms after the original transmission, the first duplicate is transmitted and 100 ms after that, the second duplicate is transmitted. In other words, the same packet is transmitted three times over a period of 150 ms.

```
v=0
o=ali 1122334455 1122334466 IN IP4 red.example.com
s=Temporal Interleaving
t=0 0
m=video 30000 RTP/AVP 100
c=IN IP4 233.252.0.1/127
a=source-filter:incl IN IP4 233.252.0.1 198.51.100.1
a=rtpmap:100 MP2T/90000
a=interleaving-period:50:100
a=mid:S2
```

5. Performance Evaluation and Reporting

Editor's note: This section should discuss how the receiver should prepare the RTCP receiver reports or whether a new XR report is needed.

6. Security Considerations

The 'interleaving-period' attribute is not believed to introduce any significant security risk to multimedia applications. A malevolent third party could use this attribute to misguide the receiver(s) about the interleaving periods and/or the number of redundant streams. For example, if the malevolent third party increases the value of the interleaving period, the receiver(s) will unnecessarily incur a longer delay since they will have to wait for the entire interleaving period. Or, if the interleaving period is reduced by the malevolent third party, the receiver(s) might not wait long enough for the duplicated transmission and incur unnecessary packet losses. However, these require intercepting and rewriting the packets carrying the SDP description; and if an interceptor can do that, many more attacks are also possible.

In order to avoid attacks of this sort, the SDP description needs to be integrity protected and provided with source authentication. This can, for example, be achieved on an end-to-end basis using S/MIME [RFC5652] [RFC5751] when SDP is used in a signaling packet using MIME types (application/sdp). Alternatively, HTTPS [RFC2818] or the authentication method in the Session Announcement Protocol (SAP) [RFC2974] could be used as well.

7. IANA Considerations

The following contact information shall be used for all registrations in this document:

Ali Begen
abegen@cisco.com

Note to the RFC Editor: In the following, replace "XXXX" with the number of this document prior to publication as an RFC.

7.1. Registration of SDP Attributes

This document registers a new attribute name in SDP.

SDP Attribute ("att-field"):	
Attribute name:	interleaving-period
Long form:	Interleaving period for temporally redundant streams
Type of name:	att-field
Type of attribute:	Media or session level
Subject to charset:	No
Purpose:	Specifies the interleaving period(s) for redundant stream(s)
Reference:	[RFCXXXX]
Values:	See [RFCXXXX]

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

8.2. Informative References

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[I-D.ietf-fecframe-framework] Watson, M., Begen, A., and V. Roca, "Forward Error Correction (FEC) Framework", draft-ietf-fecframe-framework-14 (work in progress),

March 2011.

[RFC2198] Perkins, C., Kouvelas, I., Hodson, O., Hardman, V., Handley, M., Bolot, J., Vega-Garcia, A., and S. Fosse-Parisis, "RTP Payload for Redundant Audio Data", RFC 2198, September 1997.

[IC2011] Evans, J., Begen, A., Greengrass, J., and C. Filsfils, "Towards Lossless Video Transport (in submission to IEEE Internet Computing)", 2011.

[RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, September 2009.

[RFC5751] Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", RFC 5751, January 2010.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[RFC2974] Handley, M., Perkins, C., and E. Whelan, "Session Announcement Protocol", RFC 2974, October 2000.

Authors' Addresses

Ali Begen
Cisco
181 Bay Street
Toronto, ON M5J 2T3
Canada

Email: abegen@cisco.com

Yiqun Cai
Cisco
170 W. Tasman Dr.
San Jose, CA 95134
USA

Email: ycai@cisco.com

Heidi Ou
Cisco
170 W. Tasman Dr.
San Jose, CA 95134
USA

Email: hou@cisco.com

DISPATCH WG
Internet-Draft
Intended status: Informational
Expires: August 8, 2011

J. Elwell
A. Hutton
Siemens Enterprise Communications

February 4, 2011

ICE Updated Offer Problematic
draft-elwell-mmusic-ice-updated-offer-00.txt

Abstract

Interactive Connectivity Establishment (ICE) requires an updated offer-answer cycle under some circumstances, to satisfy the needs of some devices on the signalling path (middleboxes). When used with SIP, this additional offer-answer cycle interacts with other things, such as fax and third party call control (3PCC). This document describes the problems and discusses possible remedies.

This work is being discussed on the dispatch@ietf.org mailing list.

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 August 8, 2011.

Copyright Notice

Table of Contents

1. Introduction

Interactive Connectivity Establishment (ICE) [RFC5245] specifies a mechanism for NAT traversal for multimedia sessions established using the Session Description Protocol (SDP) [RFC4566] offer-answer model [RFC3264]. It allows a pair of endpoints to exchange candidate IP addresses and ports, perform checks to see which pairs of candidates work, and agree which pairs to use for a given component of a given medium (e.g., RTP stream, RTCP stream). The mechanism can also be used for IPv6 transition, for determining whether to use IPv4 or IPv6. A particular application of ICE is with the Session Initiation Protocol (SIP) [RFC3261].

Connectivity checks are performed on the media path between candidate pairs. Based on the results of connectivity checks and certain rules, the two endpoints each determine which pair of candidates to use for a given component and can then start exchanging data (e.g., RTP packets) on the agreed path. Further exchanges on the signalling path (i.e., the path on which the offer-answer exchange is performed) are not necessary for the endpoints to agree which candidates to use.

However, certain devices on the signalling path need to know which candidates have been selected (e.g., to prioritize that traffic or to remove the resources for non-selected candidates). For this reason ICE mandates a further offer-answer exchange in some circumstances, i.e., an updated SDP offer followed by an updated SDP answer. In some situations with SIP, this updated offer-answer exchange can be problematic. This document examines these problems.

2. Fax calls

2.1. Problem statement

Except where dedicated fax devices are involved, fax calls typically start as audio. Detection of CNG tone (calling tone) from the initiating fax machine and CED (called) tone from the receiving fax machine initiates a switch to T.38, i.e., a switch from audio to image. Where the audio call uses a compressed codec (e.g., G.729), if one tone is detected there may first be a switch to G.711, for more reliable tone detection or in case the call turns out to be a non-fax modem call. Thus there can be:

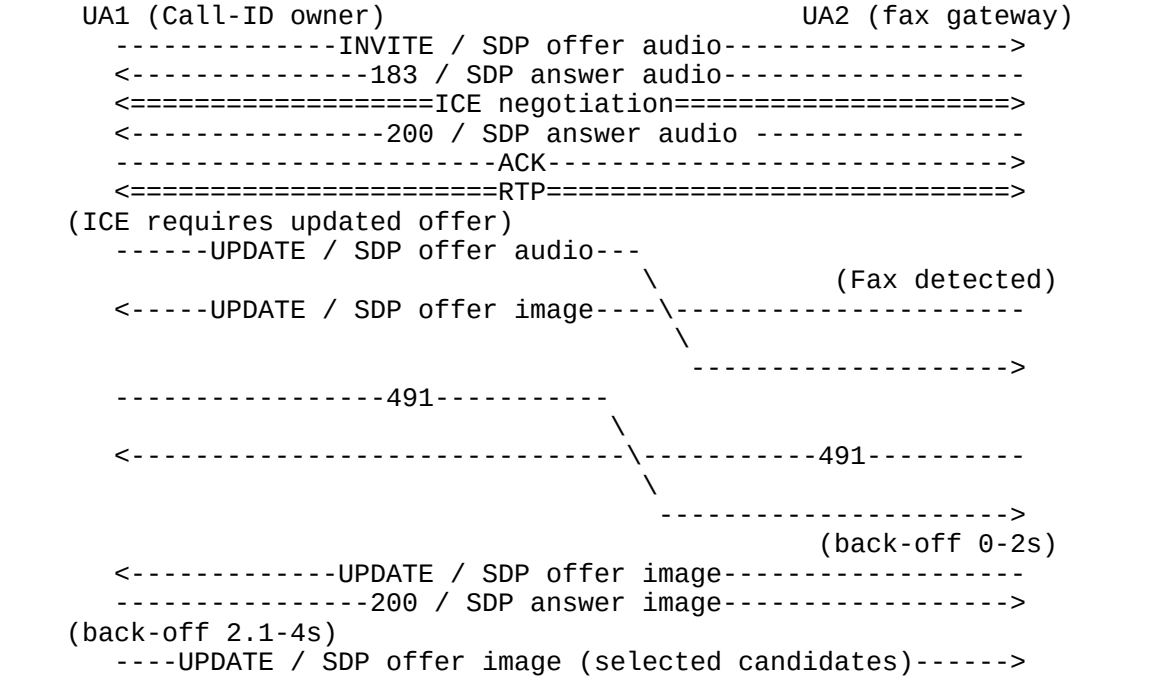
a switch from a compressed codec to G.711; or

a switch from audio to image; or

both in sequence.

Switching codec or switching from audio to image requires an SDP offer-answer cycle. ICE also requires an updated offer-answer cycle where the selected candidates are not those in the m/c-lines of the original offer-answer. If the UA that detects the need to switch because of fax is also the controlling agent from the ICE perspective, updated offer-answer for fax can follow the updated offer-answer for ICE and probably won't lead to problems.

However, if the UA that detects the need to switch because of fax is not the controlling agent from the ICE perspective, there is a significant danger of the two re-INVITE or UPDATE [RFC3311] requests colliding, resulting in a 491 response to each. According to [RFC3261] and [RFC3311], one UA (the one that owns the Call-ID) backs off for between 2.1 and 4 seconds, and the other UA backs off for between 0 and 2 seconds, before trying again. This can result in a delay of up to 4 seconds before the switch to fax, long enough in practice to cause fax calls to fail. It can also result in a delay of up to 4 seconds before the post-ICE updated offer-answer. Middleboxes that need the post-ICE updated offer-answer might not permit the flow of RTP packets throughout this period, which might also lead to failure of the fax call. An example flow is shown below:



<----200 / SDP answer image (selected candidates)--------

In this example UA1 is the ICE controlling agent and issues an updated offer on completion of ICE, and UA2 is a fax gateway that detects fax and attempts to change to image. UPDATE is supported by both and used for the updated offers. UA1 owns the Call-ID and has the longer back-off. In this example, the switch to image will probably be accomplished fast enough (back-off does not exceed 2 seconds), but the post-ICE updated offer can be delayed up to 4 seconds, perhaps leading to undesirable middlebox behaviour that might disrupt the flow of RTP and cause the fax call to fail.

Of course, collision of UPDATE or re-INVITE requests will not always occur - it is matter of timing. However, the probability of collision is not insignificant and, if that occurs, the probability of the fax call being adversely affected to the extent that it fails is not insignificant.

2.2. Possible remedies

2.2.1. Delay the ICE updated offer

UA1, as the ICE controlling agent, will be unaware that UA2 will detect fax. Therefore any delay in sending the ICE updated offer will need to apply to all calls and will need to be long enough to allow for differing amounts of time for UA2 to detect fax (perhaps several seconds). The question then is whether this would be long enough to introduce a risk of undesirable middlebox behaviour, which could impact all calls, not just fax calls.

2.2.2. Delay the fax updated offer

UA2 will know that ICE has been used, and therefore can expect an updated offer from UA1, the ICE controlling agent. Normally this should arrive quite quickly (e.g., well under 100 ms), although it depends on the number of SIP intermediaries on the path and whether any retransmissions are needed because of packet loss. Therefore a delay of a 100 ms., say, would probably not impact the fax call and would help avoid collisions but would not be a guarantee.

3. Third party call control (3PCC)

3.1. Problem statement

3PCC [RFC3725] is a common technique used with SIP where calls are controlled from an application at a SIP B2BUA. In particular, calls can be established by 3PCC, whereby the application first makes a

call to the first party (typically the device of a user requesting the call) and then makes a second call to the second party, the two calls being joined together such that media flow directly between the two devices. A typical implementation is in accordance with Flow I in [RFC3725]: the controlling B2BUA sends an offerless INVITE request to UA1, which alerts the first user. When the user answers, UA1 sends an offer in a 200 response to the INVITE request, and this offer is used by the B2BUA in a second INVITE request, this time to UA2.

The problem with using ICE with 3PCC is that 3PCC signalling does not permit the updated offer-answer for ICE to occur in a timely manner. UA2 will often take some time (seconds or tens of seconds) before sending the 200 response to its INVITE request. Yet if UA2 has already sent an SDP answer (e.g., in a 183 response), ICE can complete on the media paths and UA1, as the ICE controlling agent, can attempt an updated offer. This updated offer cannot be forwarded to UA2 until the INVITE transaction on that leg of the call has completed.

More specifically, consider the following example flow:

```
  UA1 (Call-ID owner)         B2BUA                          UA2
    <----INVITE (no SDP)------
    -----200 / SDP offer----->     ----INVITE / SDP offer--->
    <----ACK / SDP answer-----     <-----183 / SDP answer----
    <==================ICE negotiation====================>
(ICE requires updated offer)
    ----UPDATE / SDP offer---> What next?
```

In this case, UA2 sends a 183 provisional response to its INVITE request. This contains an SDP answer, which is passed to UA1 through the ACK request. Thus UA1 and UA2 are able to conduct ICE negotiation on the media paths. UA2 will probably not alert its user until ICE negotiation is complete, but anyway, there will often be a significant delay before the user answers and UA2 sends a 200 response to its INVITE request. Meanwhile, UA1, as the ICE controlling agent, attempts to send an updated offer. In this case it chooses to use an UPDATE request, but similar considerations apply if it uses a re-INVITE request. The B2BUA cannot pass that request on until the INVITE transaction with UA2 has completed. Either the UPDATE request has to be delayed somehow or rejected, in either case leading to the possibility of undesirable middlebox behaviour. For example, UA2 might be transmitting early media, which middleboxes fail to pass through correctly as a result of not receiving a timely updated offer, or clipping may occur when the user answers.

3.2. Possible remedies

3.2.1. Avoid 3PCC

There are alternatives to this form of 3PCC. For example, UA1 could be instructed to issue a conventional INVITE request by sending a SIP REFER request to UA1, or by some non-SIP means. However, using a REFER request is not an option for some types of UA, for example PSTN gateways. If user 1 is a PSTN user, it is necessary to make a PSTN call to the user, and this can be achieved by sending an INVITE request to UA1, but not by sending a REFER request to UA1. Non-SIP means are either not standardized or little deployed.

A particular example of an application that uses 3PCC is one where the user uses a web page to make the call, having selected in advance the device he/she wishes to use to make the call. The application causes the B2BUA to send an INVITE request to that selected device, followed by an INVITE request to the called destination. If the selected device is, for example, a cellular device reachable via PSTN, that initial INVITE request will be sent to a PSTN gateway.

Because of the difficulties supporting such applications by other means, 3PCC is a commonly deployed technique. It is not possible to scrap 3PCC in order to introduce ICE.

3.2.2. Delay the updated offer

UA1 will typically not be aware of the state of the INVITE transaction to UA2, and will issue the updated offer in an UPDATE or re-INVITE request without knowing whether the B2BUA can pass it on. Therefore the onus is on the B2BUA to handle the situation when it receives the UPDATE or re-INVITE request. As a non-INVITE transaction, an UPDATE request has a relatively short timeout, but one possibility would be for the B2BUA to reject it with a 500 response and a Retry-After header field, relying on UA1 to try again later. In the case of re-INVITE, the B2BUA could delay forwarding the request to UA2 until the original transaction is complete. However, in either case, middleboxes between the B2BUA and UA2 will not see the updated offer in a timely manner, and therefore might fail to handle early media correctly or clip media for a short time after the call is answered.

3.2.3. Delay ICE until UA2 answers

UA2 could delay ICE until UA2 answers, which means it would not need to send SDP answer in a provisional response but could wait for the 200 response. This would mean the user would answer and experience a delay (clipping) before ICE completes and media start to flow. Since

UA2 would not be aware of the 3PCC situation, this would impact all calls to UA2, not just those that use 3PCC.

3.2.4. Issue an early 200 response to the INVITE request to UA2

UA2 could issue a 200 response instead of a 183 response, even though the user has not yet been alerted and answered. This would be different from normal practice and might adversely impact behaviour at other SIP entites, e.g., charging, logging, forking, call forwarding. Again UA 2 would not be aware of the 3PCC situation, so this would impact all calls to UA2, not just those that use 3PCC.

3.2.5. Use reliable provisional responses

If UA2 and the B2BUA support reliable provisional responses [RFC3262], UA2 can send the 183 response with SDP answer reliably (resulting in a PRACK transaction), and then the B2BUA can send an UPDATE request with the updated offer without waiting for the INVITE transaction to complete. This would seem to work, except that it requires the entities involved to support [RFC3262]. [RFC3262] is known to be rather complicated to implement (hence the reason the ICE mechanism was specifically designed to allow SDP answer to be sent in an unreliable provisional response (ICE provides acknowledgement on the media path, rather than requiring the use of PRACK). Therefore ICE implementations should not be expected to support [RFC3262]. In particular, UA2, which is the "innocent party" in 3PCC, should not be expected to provide special functionality just to make 3PCC work.

4. Conclusions

This document illustrates two common use cases where the introduction of ICE can lead to problems with the updated offer/answer cycle that ICE requires in certain circumstances. In the first case (fax), the problem arises at the two endpoints that are trying to accomplish ICE. In the second case (3PCC), the problem arises because of a particular B2BUA behaviour, yet the B2BUA is not involved in ICE and should not need to know anything about ICE. In both cases there are work-arounds, but these introduce dependencies that contrive to reduce the chances of successful interoperability.

The need, in some circumstances, to conduct an updated offer/answer cycle on conclusion of ICE is common to both problems. This need arises not from ICE itself, but from the certain types of middlebox whose normal functioning is impacted when endpoints use ICE, unless the middleboxes have been upgraded to cope with the effects of ICE.

The two use cases illustrated might not be the only cases where the

ICE updated offer is problematic. As more complex multimedia situations arise, involving mid-call (and in particular early-in-the-call) offer-answer cycles for changing media, changing security, etc., the more likely it is that the additional ICE update offer-answer cycle will intrude in an unhelpful way.

Consequently it would be beneficial to find a way of eliminating the need for the updated offer-answer cycle.

5. IANA considerations

This document requires no IANA actions.

6. Security considerations

This document does not introduce any new security considerations.

7. Informative References

[RFC3261] 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.

[RFC3262] Rosenberg, J. and H. Schulzrinne, "Reliability of Provisional Responses in Session Initiation Protocol (SIP)", RFC 3262, June 2002.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

[RFC3311] Rosenberg, J., "The Session Initiation Protocol (SIP) UPDATE Method", RFC 3311, October 2002.

[RFC3725] Rosenberg, J., Peterson, J., Schulzrinne, H., and G. Camarillo, "Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP)", BCP 85, RFC 3725, April 2004.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT)

Traversal for Offer/Answer Protocols", RFC 5245,
April 2010.

Authors' Addresses

John Elwell
Siemens Enterprise Communications

Phone: +44 1908 817801
Email: john.elwell@siemens-enterprise.com

Andy Hutton
Siemens Enterprise Communications

Phone: +44 1908 817920
Email: andrew.hutton@siemens-enterprise.com

mmusic
Internet-Draft
Intended status: Standards Track
Expires: August 29, 2011

B. Greevenbosch
Y. Hui
Huawei Technologies

February 25, 2011

SDP attribute to signal parallax
draft-greevenbosch-mmusic-parallax-attribute-00

Abstract

This document introduces a "ParallaxInfo" attribute in SDP. This attribute can be used in stereoscopic applications, to signal the depth positioning of 2D media data within the 3D space.

Note

Discussion and suggestions for improvement are requested, and should be sent to mmusic@ietf.org.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

To see a 3D scene, the human brain interprets two different views as perceived by the left and right eyes, and fuses these views into a single 3D perception. The depth of the object is perceived by interpreting the horizontal shift of that object between the views. This shift is called "parallax".

In stereoscopic 3D multimedia applications, there are various ways to transmit media streams in 3D. One way is to transmit two different streams, one for the left eye and one for the right eye. These streams are then projected to the relevant eyes using the appropriate technology.

When sending text streams in 3D, the solution mentioned above would imply sending the same text stream twice. Since the two streams would only differ in horizontal positioning, this introduces a lot of unnecessary overhead.

This document specifies a "ParallaxInfo" attribute in SDP [RFC4566], which is used to transfer the parallax information. It eliminates the need to send two different streams separately, as they can be calculated from a single stream and the "ParallaxInfo" attribute value.

The attribute transfers this information as two parameters: one indicating which view (left/right/centre) is carried by the stream, and another to signal the parallax between the objects.

The attribute is not restricted for signalling the parallax of text streams, but it can also be used to place other 2D objects in the 3D space. Examples include a channel logo, an electronic programme guide and on-screen display of an audio volume indicator.

2. Requirements notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Definitions

L-view
: A visual entity that is to be projected to the left eye. In the case of video, the L-view is a video frame designated for the left eye. In the case of text, the L-view is the text positioned for viewing by the left eye.

L-stream
: A sequence of L-views, which is streamed to the device.

R-view
: A visual entity that is to be projected to the right eye. In the case of video, the R-view is a video frame designated for the right eye. In the case of text, the R-view is the text positioned for viewing by the right eye.

R-stream
: A sequence of R-views, which is streamed to the device.

C-view
: The centre view: a visual entity as seen from a viewpoint between the left and right eyes. The C-view can be used to calculate the L- and R-views.

C-stream
: A sequence of C-views, which is streamed to the device.

stereoscopic device
: A device that is able to produce and display different images for the left and right eyes, such that the viewer can experience a 3D view.

2D device
: A device that is not able to produce and display different images for the left and right eyes.

2D media stream
: A sequence of two dimensional visual entities (such as text or 2D graphics), which is streamed to the device.

4. The ParallaxInfo attribute

The SDP attribute "ParallaxInfo" is used to transmit the depth positioning of 2D media data, such as a text stream, in the 3D space.

The attribute has the following syntax:

a=ParallaxInfo:<transmitted position> <parallax>

The <transmitted position> indicates whether the L-, C- or R-stream is transmitted, whereas <parallax> indicates the parallax (i.e. shift) between corresponding L- and R-views in pixels.

The <transmitted position> can have the following values:

"L" indicates that the transmitted stream is the L-stream. A stereoscopic device MUST calculate the corresponding R-views by shifting the L-views <parallax> pixels towards the right.

"C" indicates that the transmitted stream is the C-stream. A stereoscopic device MUST calculate the corresponding L-views by shifting the C-views <parallax>/2 pixels towards the left, and the R-views by shifting the C-views <parallax>/2 pixels towards the right. <parallax> SHOULD be even. If it is odd, the divided value MUST be rounded off towards zero.

"R" indicates that the transmitted stream is the R-stream. A stereoscopic device MUST calculate the corresponding L-views by shifting the R-views <parallax> pixels towards the left.

<parallax> MAY be negative. In this case, the shift direction is reversed.

The "ParallaxInfo" attribute can be a session-level attribute or a media-level attribute. As a session-level attribute, it specifies the default parallax value which can be applied to all the 2D media streams in the session being described. As a media-level attribute, it specifies the parallax value which can be applied to the associated 2D media stream, overriding any session-level parallax value specified.

The stereoscopic device MAY use the session-level attribute value for on-screen display, for example an audio volume indicaton, channel indication or electronic programme guide.

Notice that a 2D device that does not support the "ParallaxInfo" attribute will ignore it, and therefore display the 2D media data on the position as transmitted.

5. Example

The following is an example of an SDP description of a session with an audio stream, a video stream and a 3GPP timed text stream (see [3gpp-tt]), streamed using RTP as per [RFC4396]. The 3GPP timed text stream is the L-stream, and each R-view is 16 pixels on the left of the L-view. Notice that this corresponds to the virtual positioning of the text in front of the screen.

```
v=0
o=Alice 2890844526 2890842807 IN IP4 131.163.72.4
s=The technology of 3D-TV
c=IN IP4 131.164.74.2
t=0 0
a=ParallaxInfo:L -20
m=video 49170 RTP/AVP 99
a=rtpmap:99 H264/90000
m=video 52888 RTP/AVP 97
a=rtpmap:97 3gpp-tt/1000
a=ParallaxInfo:L -16
m=audio 52890 RTP/AVP 10
a=rtpmap:10 L16/16000/2
```

Notice that the default value "-20" is overridden by the value "-16" for the text stream. However the "-20" value is still signalled for on-screen display of e.g. volume control and other 2D graphics.

In case each R-view is 24 pixels on the right of the associated L-view, i.e. the virtual positioning of the text is behind the screen, then the three lines defined for 3gpp-tt can be replaced as follows:

```
m=video 52888 RTP/AVP 97
a=rtpmap:97 3gpp-tt/1000
a=ParallaxInfo:L 24
```

6. Remarks

A positive parallax value indicates virtual positioning of the 2D media data behind the screen. This is the case when the objects in the L-view are on the left of the same objects in the R-view. Similarly, a negative parallax value indicates that the objects in the R-view are on the left of the same objects in the L-view, and corresponds to virtual positioning in front of the screen.

Since the "ParallaxInfo" attribute indicates a shift of the transmitted stream, it might happen that the L- or R-view trespasses the bounderies of the display. In this case, clipping is necessary, as illustrated by Figure 1.

Similarly, there are areas which are covered by the L-view but not by the R-view and vice versa. These areas need to be filled in a sensible way. Since the "ParallaxInfo" attribute is designed for objects that overlay other video data (e.g. subtitles), it is trivial to fill in uncovered areas by using the underlying video data. However, if there is no underlying video data, other mechanisms to fill in the uncovered areas need to be defined. Definition of these mechanisms are out of scope of this document.

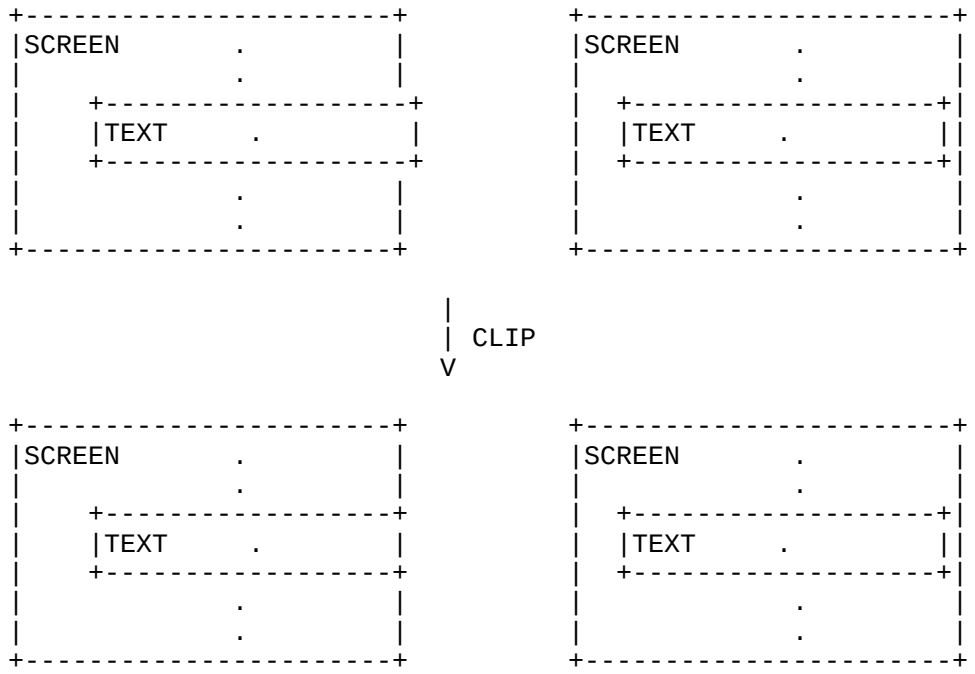


Figure 1

7. Security Considerations

The authors foresee no security issues in addition to those already listed in [RFC4566].

8. IANA Considerations

Following the guidelines in [RFC4566], the SDP attribute has to be registered at IANA:

- Contact name/email: authors of this RFC
- Attribute name: ParallaxInfo
- Long-form attribute name: Parallax info for the depth positioning of 2D media data in the 3D space
- Type of attribute: session level and media level
- Subject to charset: no

This attribute is used to signal how 2D media data is to be displayed in the 3D space. It indicates the shift of the respective left and right views.

The attribute has the following ABNF (see [RFC4234]) description:

```
ParalaxInfo          = "a=ParallaxInfo:" TransmittedPosition Parallax
TransmittedPosition  = "C"/"L"/"R"
Parallax             = num-val
```

The attribute transfers this information as two parameters: "TransmittedPosition" indicates which view of the 2D media data (left "L", right "R" or centre "C") is carried by the stream, and "Parallax" signals the parallax (in pixels) of objects in the 2D media stream.

9. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.

[RFC4396] Rey, J. and Y. Matsui, "RTP Payload Format for 3rd Generation Partnership Project (3GPP) Timed Text", RFC 4396, February 2006.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[3gpp-tt] 3GPP, "Transparent end-to-end packet switched streaming service (PSS); Protocols and codecs (Release 5)", TS 26.234 v5.3.0, December 2002.

Authors' Addresses

Bert Greevenbosch
Huawei Technologies Co., Ltd.
Huawei Industrial Base
Bantian, Longgang District
Shenzhen 518129
P.R. China

Phone: +86-755-28978088
Email: bgreeven@huawei.com

Hui Yu
Huawei Technologies Co., Ltd.
Huawei Nanjing R&D Center
101 Software Avenue
Yuhuatai District
Nanjing 210012
P.R. China

Phone: +86-25-56620323
Email: huiyu@huawei.com

mmusic
Internet-Draft
Intended status: Standards Track
Expires: September 4, 2011

B. Greevenbosch
Y. Hui
Huawei

March 3, 2011

Signal 3D format
draft-greevenbosch-mmusic-signal-3d-format-00

Abstract

This document introduces the SDP attribute "3dFormat", which provides format description of stereoscopic 3D video. In addition, the grouping mechanism for SDP is extended to cater for stereoscopic 3D video.

Note

Discussion and suggestions for improvement are requested, and should be sent to mmusic@ietf.org.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

In stereoscopic 3D multimedia applications, two views are displayed, one for the left eye and one for the right eye.

There are various ways of formatting the views of Stereoscopic 3D video. Examples of 3D formats are frame packing (see [HDMIv1.4a]) and the combination of 2D video and auxiliary data such as depth maps (see [ISO/IEC 23002-3]). Stereoscopic 3D video may be carried over a single stream or over several streams, depending on its 3D format.

In multimedia streaming applications, the Session Description Protocol (SDP) [RFC4566] can be used to provide to the receiver sufficient information about the media streams, and to enable the receiver to join and participate in the session.

This document defines an extension to SDP that provides sufficient information about the format of stereoscopic 3D video carried in the media stream(s). Before accessing the stream(s), the receiver can use the 3D format description from SDP to determine whether it has the capability to receive and render the stereoscopic 3D video content, and whether it can participate in the session.

The mentioned SDP extension is a new SDP attribute "3dFormat", which provides the format description of stereoscopic 3D video. The design of the attribute is based on the following requirements, which are listed only for informational purposes:

- It SHALL be possible to signal that the left and right views are carried in a single stream, by the use of frame packing.
- It SHALL be possible to signal that 2D video and auxiliary video (such as depth maps) are carried in a single stream.
- It SHALL be possible to signal that the left and right views are carried in two separate streams.
- It SHALL be possible to signal that 2D video and auxiliary video (such as depth maps) are carried in separate streams.

To bind multiple video streams that carry a single stereoscopic 3D video, this document also extends the SDP grouping mechanism from [RFC5888].

2. Requirements notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Definitions

2D video
: Video that does not in itself contain depth or parallax information.

auxiliary video
: A sequence of depth or parallax maps, which are used to add depth to 2D video.

C-view
: The centre view: a visual entity as seen from a viewpoint between the left and right eyes. The C-view can be used to calculate the L- and R-views.

C-stream
: A 2D video stream consisting of a sequence of C-views.

depth map
: A two dimensional map, each pixel of which defines the depth of one or more pixels in an associated 2D video frame.

depth map stream
: An auxiliary stream, which contains a sequence of depth maps. The depth map stream is synchronised with the associated 2D video stream.

frame packing
: A format that packs the L- and R-views into a single 2D video stream. The packing may be done spatially, where each video frame is divided into sub-frames, one containing the L-view and one containing the R-view. The packing can also be done sequentially, where alternating video frames represent L- and R-views.

L-view
: A visual entity that is to be projected to the left eye.

L-stream
: A 2D video stream consisting of a sequence of L-views.

parallax map
: A two dimensional map, each pixel of which defines the parallax of one or more pixels in an associated 2D video frame.

parallax map stream
: An auxiliary stream, which contains a sequence of parallax maps. The parallax map stream is synchronised with the associated 2D video stream.

R-view
A visual entity that is to be projected to the right eye.

R-stream
A 2D video stream consisting of a sequence of R-views.

stereoscopic 3D video
The L- and R-streams, ready to be projected to the viewer’s left and right eyes.

sub-frame
A part of a video frame.

4. The "3dFormat" attribute

The media-level SDP attribute "3dFormat" signals the format of stereoscopic 3D video. The attribute transfers this information through two parameters: one indicating the format type of the stereoscopic 3D video carried in the media stream(s), and the other indicating the type of the video component, which is a constituent element of the stereoscopic 3D video. The video component type depends on the format type of the stereoscopic 3D video. The syntax of the attribute is defined as follows:

a=3dFormat:<Format Type> <Component Type>

The <Format Type> can have the following values (as indicated between the quotes):

"FP" Frame Packing
The L- and R-views are packed into a single stream. The packing may use a side-by-side, top-and-bottom, interleaved, checkerboard or frame sequential format.

"SC" Simulcast
The L- and R-streams are transmitted separately.

"2DA" 2D + auxiliary
2D video and auxiliary data (such as depth maps or parallax maps) are transmitted. These can be transmitted in a single stream, as well as in two separate streams.

The <Component Type> can have the following values (as indicated between the quotes):

"C" Centre view
The associated stream is a C-stream.

"CD" centre view and depth map
The associated stream contains both the C-view and depth map sequences.

"ChB" Checkerboard
The video frame consists of alternating pixels from the corresponding L- and R-views, as illustrated by Figure 1.

"CP" Centre view and parallax map
The associated stream contains both the C-view and parallax map sequences.

"D" Depth map
The associated stream is a sequence of depth maps.

"L" Left view
The associated stream is the L-stream.

"LD" Left view and depth map
The associated stream contains both the L-view and depth map sequences.

"LIL" Line Interleaved
Each video frame consists of alternating scan lines from the L- and R-views.

"LP" Left view and parallax map
The associated stream contains both the L-view and parallax map sequences.

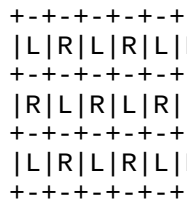
"P" Parallax map
The associated stream is a sequence of parallax maps.

"R" Right view
The associated stream is the R-stream.

"SbS" Side by Side
Each video frame is divided in two equally sized sub-frames, spatially positioned side by side of each other. One sub-frame contains the L-view, whereas the other contains the R-view.

"Seq" Frame sequential
The single video stream consists of alternating frames from the L- and R-streams.

"TaB" Top and Bottom
Each video frame is divided in two equally sized sub-frames, spatially positioned above each other. One sub-frame contains the L-view, whereas the other contains the R-view.



The checkerboard pattern. The transmitted video frame is composed of pixels from the L- and R-views. Samples from the L-view are indicated with "L", whereas samples from the R-view are indicated with "R".

Figure 1

5. Grouping

When multiple streams carry a single stereoscopic 3D video, (e.g. C-stream and parallax map, or separately transmitted L- and R-streams), the grouping mechanism from [RFC5888] MUST be used.

However, to cater for the special requirements of 3D signalling, the semantics are expanded:

```
group-attribute     = "a=group:" semantics *(SP identification-tag)
semantics           = "LS" / "FID" / "3DS" / semantics-extension
semantics-extension = token
```

The grouping is needed when multiple streams carry a single stereoscopic 3D video. This is the case when the <format type> is "SC", or the <format type> is "2DA" and the 2D video and auxiliary data are transmitted as multiple streams. A group with the "3Ds" semantics is called a "3DS group".

A 3DS group MUST NOT contain data that is (potentially) inconsistent with other data in the 3DS group:

- A 3DS group MUST NOT contain both a parallax map stream and a depth map stream.
- A 3DS group MUST NOT contain more than one parallax map stream.
- A 3DS group MUST NOT contain more than one depth map stream.
- A 3DS group MUST contain at least one 2D video stream.
- If a 3GS group contains an L- and an R-stream, it MUST NOT contain a depth map or a parallax map.
- If a 3DS group contains only one 2D video stream, it MUST also contain a parallax map stream or a depth map stream.
- If a 3DS group contains a parallax map stream or a depth map stream, it MUST also contain a 2D video stream.

6. Combinations of attribute values and group usage

The following table summarises the possible combinations of attribute values and grouping:

	FP	SC	2DA
C			D/P,3DS
CD			T
ChB	T		
CP			T
D			C/L,3DS
L		R,3DS	D/P,3DS
LD			T
LIL	T		
LP			T
P			C/L,3DS
R		L,3DS	
SbS	T		
Seq	T		
TaB	T		

The table is to be read as follows:

- The columns indicate <Format Type> values, whereas the rows indicate <Component Type> values.
- For one particular column, we denote the <Format Type> value by "FT" and the <Component Type> value by "CT".
- When an entry in the table is empty, it means that the corresponding combination of FT and CT is not allowed.

- When an entry in the table contains a single <Component Type> value CTsec, it means that another stream with the <Component Type> value CTsec and the same <Format Type> value FT is needed.

- When multiple <Component Type> values are listed, separated by a "/" symbol, only one secondary stream is needed, which must have one of the listed <Component Type> values, and the same <Format Type> value FT.

- When an entry contains "3DS", it means that a 3DS group is needed.

- When an entry in the table contains the letter "T" (true), it means that the corresponding combination FT and CT is allowed, that there is no required secondary stream, and that a 3DS group is not needed.

7. Examples

7.1. One single frame compatible stream

The following is an example of an SDP description of a session which contains a single stream, in which the L- and R-streams are packed, in side by side fashion.

```
v=0
o=Alice 2890844526 2890842807 IN IP4 131.163.72.4
s=The technology of 3D-TV
c=IN IP4 131.164.74.2
t=0 0
m=video 49170 RTP/AVP 99
a=rtpmap:99 H264/90000
a=3dFormat:FP SbS
m=audio 52890 RTP/AVP 10
a=rtpmap:10 L16/16000/2
```

7.2. Two separate streams

The following is an example of an SDP description of a session with an audio stream, an L-stream and an R-stream.

```
v=0
o=Alice 2890844526 2890842807 IN IP4 131.163.72.4
s=The technology of 3D-TV
c=IN IP4 131.164.74.2
t=0 0
a=group:3DS 1 2
m=video 49170 RTP/AVP 99
a=rtpmap:99 H264/90000
a=3dFormat:SC L
a=mid:1
m=video 49172 RTP/AVP 101
a=rtpmap:101 H264/90000
a=3dFormat:SC R
a=mid:2
m=audio 52890 RTP/AVP 10
a=rtpmap:10 L16/16000/2
```

7.3. C-stream and depth map stream

The following is an example of an SDP description of a session with an audio stream, a C-stream and a depth map stream.

```
v=0
o=Alice 2890844526 2890842807 IN IP4 131.163.72.4
s=The technology of 3D-TV
c=IN IP4 131.164.74.2
t=0 0
a=group:3DS 1 2
m=video 49170 RTP/AVP 99
a=rtpmap:99 H264/90000
a=3dFormat:2DA C
a=mid:1
m=video 49172 RTP/AVP 101
a=rtpmap:101 H264/90000
a=3dFormat:2DA D
a=mid:2
m=audio 52890 RTP/AVP 10
a=rtpmap:10 L16/16000/2
```

7.4. Stereoscopic 3D video with two different formats

In the following example, there are two different formats for stereoscopic 3D video. One consists of stream 1 (C-stream) and stream 2 (parallax map stream), whereas the other consists of stream 3 (L-stream) and stream 4 (R-stream). There also is an audio stream, which can be used with both formats.

```
v=0
o=Alice 2890844526 2890842807 IN IP4 131.163.72.4
s=The technology of 3D-TV
c=IN IP4 131.164.74.2
t=0 0
a=group:3DS 1 2
a=group:3DS 3 4
m=video 49170 RTP/AVP 99
a=rtpmap:99 H264/90000
a=3dFormat:2DA C
a=mid:1
m=video 49172 RTP/AVP 101
a=rtpmap:101 H264/90000
a=3dFormat:2DA P
a=mid:2
m=video 49174 RTP/AVP 103
a=rtpmap:103 H264/90000
a=3dFormat:SC L
a=mid:3
m=video 49176 RTP/AVP 105
a=rtpmap:105 H264/90000
a=3dFormat:SC R
a=mid:4
m=audio 52890 RTP/AVP 10
a=rtpmap:10 L16/16000/2
```

8. Formal ABNF grammar of the "3dFormat" attribute

This section contains the formal ABNF grammar of the "3dFormat" attribute.

```
3dFormat-attribute = "a=3dFormat:" formatType componentType
formatType         = "FP"/"SC"/"2DA"
componentType      = "C"/"CD"/"ChB"/"CP"/"D"/"L"/"LD"/
                     "LIL"/"LP"/"P"/"R"/"SbS"/"Seq"/"TaB"
```

9. Security Considerations

The authors foresee no security issues in addition to those already listed in [RFC4566].

10. IANA Considerations

10.1. "3dFormat" attribute

Following the guidelines in [RFC4566], the SDP attribute has to be registered at IANA:

- Contact name/email: authors of this RFC
- Attribute name: 3dFormat
- Long-form attribute name: Attribute for signalling the format of a stereoscopic 3D video carried in the media stream(s).
- Type of attribute: media level
- Subject to charset: no

The "3dFormat" SDP media-level attribute is used to signal the format of stereoscopic 3D video, carried in one or more media stream(s).

The attribute has the following syntax:

a=3dFormat:<Format Type> <Component Type>

The <Format Type> indicates the format type of the stereoscopic 3D video carried in the media stream(s). It indicates whether the stereoscopic 3D video is frame packed, simulcast or consists of a 2D video stream and an auxiliary stream. The <Format Type> can have the following values (as indicated between the quotes):

"FP"	frame packed
"SC"	simulcast
"2DA"	2D + auxiliary

The <Component Type> indicates the type of the video component, which is a constituent element of the stereoscopic 3D video. It can have the following values:

```
"C"      centre view
"CD"     centre view and depth map
"ChB"    checkerboard
"CP"     centre view and parallax map
"D"      depth map
"L"      left view
"LD"     left view and depth map
"LIL"    line interleaved
"LP"     left view and parallax map
"P"      parallax map
"R"      right view
"SbS"    side by side
"Seq"    frame sequential
"TaB"    top and bottom
```

10.2. "3DS" value for "group" semantics

Following the standards action policy from [RFC5226], the following semantics have to be registered with IANA in the "Semantics for the "group" SDP Attribute" registry under "SDP Parameters":

Semantics	Token	Reference
3D synchronised	3DS	this RFC

11. Normative References

[HDMIv1.4a] HDMI, "HDMI Specification Version 1.4a", ISO/IEC FDIS 23002-3:2007(E), March 2010.

[ISO/IEC 23002-3] MPEG, "MPEG video technologies part 3: Representation of auxiliary video and supplemental information", ISO/IEC FDIS 23002-3:2007(E), December 2002.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework", RFC 5888, June 2010.

Authors' Addresses

Bert Greevenbosch
Huawei Technologies Co., Ltd.
Huawei Industrial Base
Bantian, Longgang District
Shenzhen 518129
P.R. China

Phone: +86-755-28978088
Email: bgreeven@huawei.com

Hui Yu
Huawei Technologies Co., Ltd.
Huawei Nanjing R&D Center
101 Software Avenue
Yuhuatai District
Nanjing 210012
P.R. China

Phone: +86-25-56620323
Email: huiyu@huawei.com

DCCP Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 10, 2011

T. Phelan
Sonus
G. Fairhurst
University of Aberdeen
March 10, 2011

Datagram Congestion Control Protocol (DCCP) Encapsulation for NAT Traversal (DCCP-UDP)
draft-ietf-dccp-udpencap-07

Abstract

This document specifies an alternative encapsulation of the Datagram Congestion Control Protocol (DCCP), referred to as DCCP-UDP. This encapsulation allows DCCP to be carried through the current generation of Network Address Translation (NAT) middleboxes without modification of those middleboxes. This document also updates the SDP information for DCCP defined in RFC 5762.

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 September 4, 2011.

Copyright Notice

Table of Contents

1. Introduction

The Datagram Congestion Control Protocol (DCCP), specified in [RFC4340], is a transport-layer protocol that provides upper layers with the ability to use non-reliable congestion-controlled flows. The current specification for DCCP [RFC4340] specifies a direct encapsulation in IPv4 or IPv6 packets.

[RFC5597] specifies how DCCP should be handled by devices that use Network Address Translation (NAT) or Network Address and Port Translation (NAPT). However, there is a significant installed base of NAT/NAPT devices that do not support [RFC5597]. In the short term, it would be useful to have an encapsulation for DCCP that is compatible with this installed base of NAT/NAPT devices that support [RFC4787], but do not support [RFC5597]. This document specifies that encapsulation, which is referred to as DCCP-UDP. For convenience, the standard encapsulation for DCCP [RFC4340] (including [RFC5596] as required) is referred to as DCCP-STD.

The encapsulation described in this document may also be used as a transition mechanism to enable support for DCCP in devices that support UDP, but do not yet natively support DCCP. This therefore also allows the DCCP transport to be implemented within an application using DCCP-UDP.

The document also updates the SDP specification for DCCP to convey the encapsulation type. In this respect only, it updates the method in [RFC5762].

The DCCP-UDP encapsulation specified in this document supports all of the features contained in DCCP-STD, but with limited functionality for partial checksums.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. DCCP-UDP

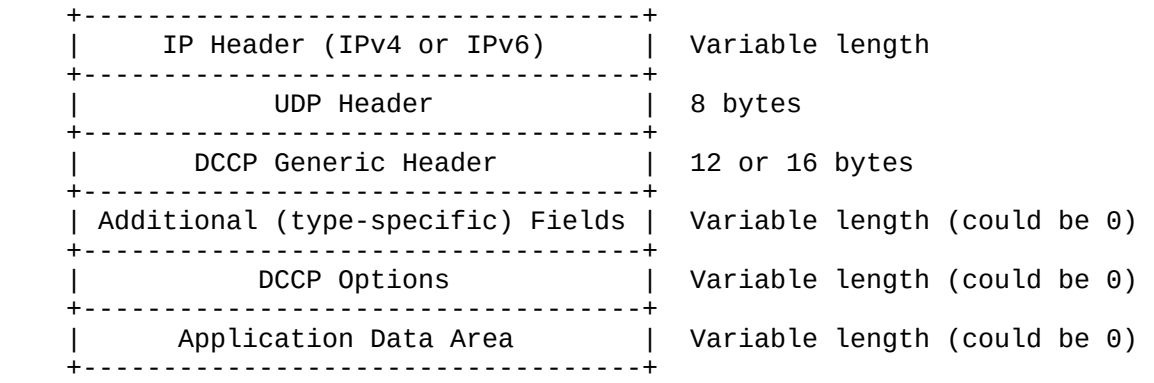
The basic approach is to insert a UDP [RFC0768] header between the IP header and the DCCP packet. Note that this is not a tunneling approach. The IP addresses of the communicating end systems are carried in the IP header. The method does not embed additional IP addresses.

The method is designed to support use when these addresses are modified by a device that implements NAT/NAPT. A NAT translates the IP addresses, which impacts the transport-layer checksum. A NAPT device may also translate the port values (usually the source port). In both cases, the outer transport header that includes these values would need to be updated by the NAT/NAPT.

A device offering or using DCCP services via DCCP-UDP encapsulation listens on a UDP port (default port, XXX IANA PORT XXX), or may bind to a specified port utilising out-of-band signalling, such as the Session Description Protocol (SDP). The DCCP-UDP server accepts incoming packets over the UDP transport and passes the received packets to the DCCP protocol module, after removing the UDP encapsulation.

A DCCP implementation MAY allow services to be simultaneously offered over any or all combinations of DCCP-STD and DCCP-UDP encapsulations with IPv4 and IPv6.

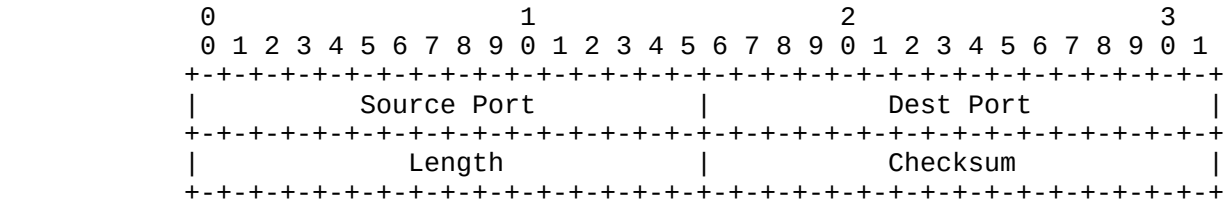
The basic format of a DCCP-UDP packet is:



Section 3.8 describes usage of UDP ports. This includes implementation of a DCCP-UDP encapsulation service as a daemon that listens on a well-known port, allowing multiplexing of different DCCP applications over the port.

3.1. The UDP Header

The format of the UDP header is specified in [RFC0768]:



For DCCP-UDP, the fields are interpreted as follows:

Source and Dest(ination) Ports: 16 bits each

These fields identify the UDP ports on which the source and destination (respectively) of the packet are listening for incoming DCCP-UDP packets. The UDP port values do not identify the DCCP source and destination ports.

Length: 16 bits

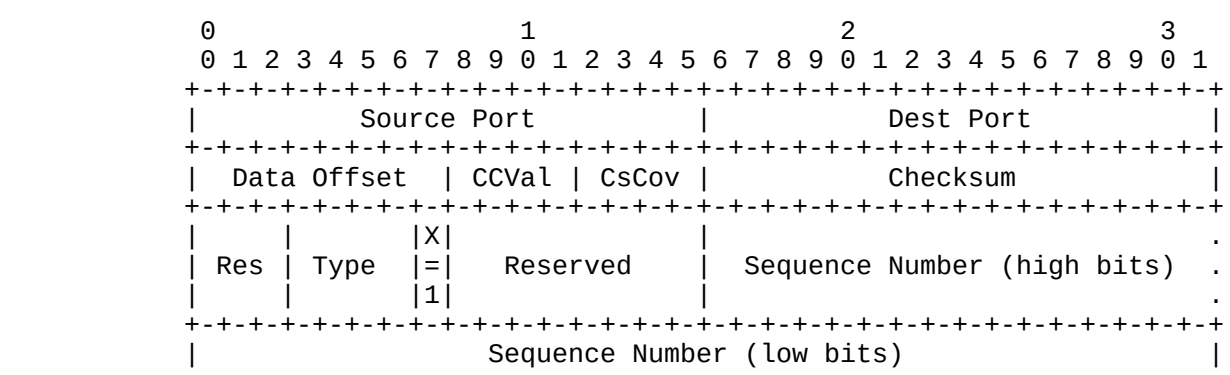
This field is the length of the UDP datagram, including the UDP header and the payload (for DCCP-UDP, the payload is a DCCP-UDP datagram).

Checksum: 16 bits

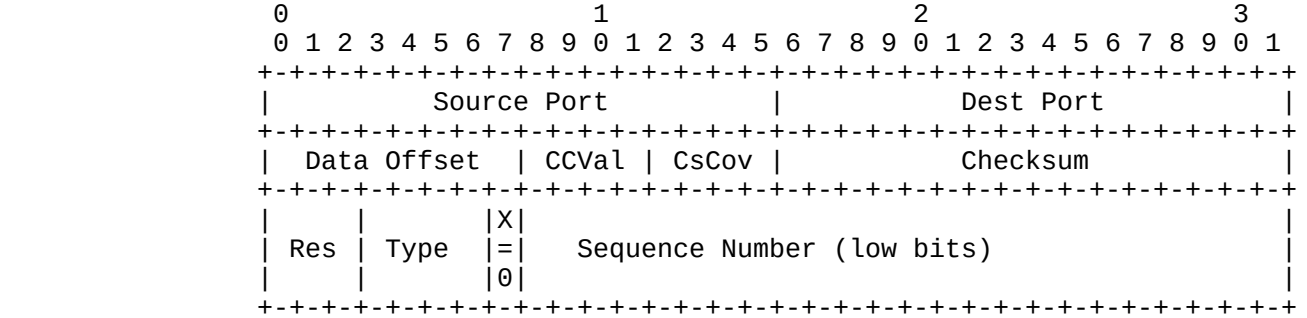
This field is the Internet checksum of a network-layer pseudoheader and Length bytes of the UDP packet [RFC0768]. The UDP checksum MUST NOT be zero for a UDP packet that carries DCCP-UDP.

3.2. The DCCP Generic Header

The DCCP Generic Header [RFC4340] takes two forms, one with long sequence numbers (48 bits) and the other with short sequence numbers (24 bits).



The Generic DCCP Header with long sequence numbers [RFC4340]



The Generic DCCP Header with short sequence numbers [RFC4340]

All generic header fields, except for the Checksum field, have the meaning specified in [RFC4340] updated by [RFC5596].

Section 3.8 describes how a DCCP-UDP implementation treats UDP and DCCP ports.

3.3. DCCP-UDP Checksum Procedures

DCCP-UDP employs a checksum at the UDP level and eliminates the use of the DCCP checksum. This approach was chosen to enable use of current NAT/NATP traversal methods developed for UDP. Such methods will generally be unaware whether DCCP is being encapsulated and hence do not update the inner checksum in the DCCP header. Standard DCCP requires protection of the DCCP header fields, this justifies any processing overhead incurred from calculating the UDP checksum.

In addition, UDP NAT traversal does not support partial checksums. Although this is still permitted end-to-end in the encapsulated DCCP datagram, links along the path will treat these as UDP packets and can not enable special partial checksum processing.

For DCCP-UDP, the function of the DCCP Checksum field is performed by the UDP checksum field. On transmit, the DCCP Checksum field SHOULD be set to zero. On receive, the DCCP Checksum field MUST be ignored.

The UDP checksum MUST NOT be zero for a UDP packet that is sent using DCCP-UDP. If the received UDP Checksum field is zero, the packet MUST be dropped.

If the UDP Length field is less than 20 (the UDP Header length and

minimum DCCP-UDP header length), the packet MUST be dropped.

If the UDP Checksum field, computed using standard UDP methods, is invalid, the packet MUST be dropped.

If the UDP Length field in a received packet is less than the length of the UDP header plus the entire DCCP-UDP header (including the generic header and type-specific fields and options, if present), or the UDP Length field is greater than the length of the packet from the beginning of the UDP header to the end of the packet, the packet MUST be dropped.

3.3.1. Partial Checksums and the Minimum Checksum Coverage Feature

This document describes an encapsulation for DCCP that uses the UDP transport. It requires the UDP checksum to be enabled. This checksum provides coverage of the entire encapsulated DCCP datagram.

DCCP-UDP supports the syntax of partial checksums. It also supports negotiation of the Minimum Checksum Coverage feature and settings of the CsCov field. However, the UDP checksum field in DCCP-UDP always covers the entire DCCP datagram and the DCCP checksum is ignored on receipt. An application that enables the partial checksums feature in the DCCP Module will therefore experience a service that is functionally identical to using full DCCP checksum coverage. This is also the service that the application would have received if it had used a network path that did not provide optimised processing for DCCP partial checksums.

3.4. Network Layer Options

A DCCP-UDP implementations MAY transfer network-layer options intended for DCCP to the network-layer header of the encapsulating UDP packet.

A DCCP-UDP endpoint that receives IP-options for the encapsulating UDP packet MAY forward these to the DCCP protocol module. If the endpoint forwards a specific network layer option to the DCCP module, it MUST also forward all subsequent packets with this option. Consistent forwarding is essential for correct operation of many end-to-end options.

3.5. Explicit Congestion Notification

A DCCP-UDP endpoint SHOULD follow the procedures of DCCP-STD section 12 by setting the ECN fields in the IP Headers of outgoing packets and examining the values received in the ECN fields of incoming IP packets, relaying any packet markings to the DCCP module.

Implementations that do not support ECN MUST follow the procedures in DCCP-STD section 12.1 with regard to implementations that are not ECN capable.

3.6. ICMP handling for messages relating to DCCP-UDP

To allow ICMP messages to be demultiplexed by the receiving endpoint, part of the original packet that resulted in the message is included in the payload of the ICMP error message. The receiving endpoint can therefore use this information to associate the ICMP error with the transport protocol instance that resulted in the ICMP message. When DCCP-UDP is used, the error message and the payload of the ICMP error message relate to the UDP transport.

DCCP-UDP endpoints SHOULD forward ICMP messages relating to a UDP packet that carries a DCCP-UDP to the DCCP module. This may imply translation of the payload of the ICMP message into a form that is recognised by the DCCP stack. [ICMP] describes precautions that are desirable before TCP acts on the receipt of an ICMP message. Similar precautions are desirable prior to forwarding by DCCP-UDP to the DCCP module.

The minimal length ICMP error message generated in response to processing a UDP Datagram only identifies the Source UDP Port and Destination UDP Port. This ICMP message does not carry sufficient information to discover the encapsulated DCCP Port values. A DCCP-UDP endpoint that supports multiple DCCP connections over the same pair of UDP ports (see section Section 3.8) may not therefore be able to associate an ICMP message with a unique DCCP-UDP connection.

3.7. Path Maximum Transmission Unit Discovery

DCCP-UDP implementations SHOULD follow DCCP-STD section 14 with regard to determining the maximum packet size and the use of Path Maximum Transmission Unit Discovery (PMTUD).

The effect of encapsulation is to incur additional datagram overhead. This will reduce the Maximum Packet Size (MPS) at the DCCP level.

3.8. Usage of the UDP port by DCCP-UDP

A DCCP-UDP server (that is, an initially passive endpoint that wishes to receive DCCP-Request packets [RFC4340] over DCCP-UDP) listens for connections on one or more UDP ports. UDP port number XXX IANA PORT XXX has been reserved as the default listening UDP port for a DCCP-UDP server. Some NAT/NAPT topologies may require using a non-default listening port.

The purpose of this IANA-assigned port is for the operating system or a framework to receive and process DCCP-UDP datagrams for delivery to the DCCP module. Because of this, the IANA-assigned port SHOULD NOT be used as the Destination UDP Port by a DCCP-UDP server listening for incoming DCCP-UDP packets and SHOULD NOT be used as a Source UDP Port by a client application sending DCCP-UDP packets.

A DCCP-UDP client provides UDP source and destination ports as well as DCCP source and destination ports at connection initiation time. A client SHOULD ensure that each DCCP connection maps to a single UDP connection by setting the UDP source port. Choosing a distinct source UDP port for each distinct DCCP connection ensures that UDP-based flow identifiers differ whenever DCCP-based flow identifiers differ. Specifically, two connections with different <source IP address, source DCCP port, destination IP address, destination DCCP port> DCCP 4-tuples will have different <source IP address, source UDP port, destination IP address, destination UDP port> UDP 4-tuples.

A DCCP-UDP server SHOULD accept datagrams from any UDP source port. There is a risk that the same DCCP source port number could be used by two endpoints each behind a NAPT. A DCCP-UDP server must therefore demultiplex a DCCP-UDP flow using both the UDP source and destination port numbers and the encapsulated DCCP ports. This ensures than an active DCCP connection is uniquely identified by the 6-tuple <source IP address, source UDP port, source DCCP port, destination IP address, destination UDP port, destination DCCP port>.

The demultiplexing at a DCCP-UDP endpoint occurs in two stages:

1) In the first stage, DCCP-UDP packets are demultiplexed using the UDP 4-tuple: <source IP address, source UDP port, destination IP address, destination UDP port>.

2) In the second stage, a receiving endpoint MUST ensure that two independent DCCP connections that were multiplexed to the same UDP 4-tuple are not associated with the same connection in the DCCP module. The endpoint therefore needs to keep state for the set of active DCCP-UDP endpoints using each combination of a UDP 4-tuple: <source IP address, source UDP port, destination IP address, destination UDP port>. A DCCP endpoint MUST implement one of the two methods:

- A DCCP server MAY accept only one active 6-tuple at any one time for a given UDP 4-tuple. In this method, DCCP-UDP packets that do not match an active 6-tuple MUST NOT be passed to the DCCP module and the DCCP Server SHOULD send a DCCP-Reset with with Reset Code XXX IANA Port Reuse XXX, "Encapsulated Port Reuse". An endpoint that receives a DCCP-Reset with this reset code will clear its

connection state, but MAY immediately try again using a different 4-tuple. This provides protection should the same UDP 4-tuple be re-used by multiple DCCP connections, ensuring that only one DCCP connection is established at one time.

- o A DCCP server MAY support multiple DCCP connections over the same UDP 4-tuple. In this method, the endpoint MUST then associate each 6-tuple with a single DCCP connection. If an endpoint is unable to demultiplex the 6-tuple (e.g. due to internal resource limits), it MUST NOT pass DCCP-UDP packets that do not match an active 6-tuple to the DCCP module. The DCCP endpoint MAY send a DCCP-Reset with Reset Code XXX IANA Port Reuse XXX, "Encapsulated Port Reuse", indicating the connection has been closed, but may be retried using a different UDP 4-tuple.

3.9. Service Codes and the DCCP Port Registry

This section clarifies the usage of DCCP Service Codes and the registration of server ports by DCCP-UDP. The section is not intended to update the procedures for allocating Service Codes or server ports.

There is one Service Code registry and one DCCP port registration that apply to all combinations of encapsulation and IP version. A DCCP Service Code specifies an application using DCCP regardless of the combination of DCCP encapsulation and IP version. An application may choose not to support some combinations of encapsulation and IP version, but its Service Code will remain registered for those combinations and the Service Code must not be used by other applications. An application should not register different Service Codes for different combinations of encapsulation and IP version. [RFC5595] provides additional information about DCCP Service Codes.

Similarly, a port registration is applicable to all combinations of encapsulation and IP version. Again, an application may choose not to support some combinations of encapsulation and IP version on its registered port, although the port will remain registered for those combinations. Applications should not register different ports just for the purpose of using different combinations of encapsulation.

4. DCCP-UDP and Higher-Layer Protocols

The encapsulation of a higher-layer protocol within DCCP MUST be the same for both DCCP-STD and DCCP-UDP. Encapsulations of DTLS over DCCP is defined in [RFC5238] and RTP over DCCP is defined in [RFC5762]. This document therefore does not update these encapsulations when using DCCP-UDP.

5. Signaling the Use of DCCP-UDP

Applications often signal transport connection parameters through outside means, such as SDP. Applications that define such methods for DCCP MUST define how the DCCP encapsulation is chosen, and MUST allow either encapsulation to be signaled. Where DCCP-STD and DCCP-UDP are both supported, DCCP-STD SHOULD be preferred.

Procedures for handling DCCP-STD and/or DCCP-UDP with Interactive Connectivity Establishment (ICE) may need to be developed, but are left for further work.

5.1. SDP support for DCCP-UDP

[RFC5762] defines SDP extensions for signaling RTP over DCCP connections. Since it predates this document, it does not define a method for determining the DCCP encapsulation type. This document updates [RFC5762] to add a method for determining the DCCP encapsulation type.

A new SDP attribute "dccp-encap" is defined for signaling the DCCP encapsulation according to the following ABNF [RFC5234]:

```
dccp-encap-attr = %x61 "=dccp-in-udp" [":" udp-port-num]

udp-port-num    = *DIGIT
```

where *DIGIT is as defined in [RFC5234].

The presence of "a=dccp-in-udp" in an SDP offer indicates that the offerer is listening for DCCP-UDP connections on the indicated UDP port (if udp-port-num is included) or on the default port for the DCCP-UDP service if no port is included. This attribute MAY also be used in a declarative SDP file.

The absence of "a=dccp-in-udp" in an SDP offer indicates that the offerer is listening for DCCP-STD connections. The presence of "a=dccp-in-udp" conveys no information about whether or not the offerer is listening for DCCP-STD connections.

The new SDP attribute specified in this section is expected to be useful when the offering party is on the public Internet, or in the same private addressing realm as the answering party. In this case, the DCCP-UDP server has a public address. The client may either have a public address or be behind a NAT/NAPT. This is considered a scenario that has the potential to be an important use-case.

Some other NAT/NAPT topologies may result in the advertised port

being unreachable via the NAT/NAPT.

5.1.1. Example of SDP use

The text below provides an example of SDP signalling, where an application signals support for both native DCCP and for DCCP-UDP:

```
v=0
o=alice 1129377363 1 IN IP4 192.0.2.47
s=-
c=IN IP4 192.0.2.47
t=0 0
m=video 5004 DCCP/RTP/AVP 99
a=rtcp-mux
a=rtpmap:99 h261/90000
a=dccp-service-code:SC=x5254505
a=dccp-in-udp:9999
a=setup:passive
a=connection:new
```

6. Security Considerations

DCCP-UDP provides all of the security risk-mitigation measures present in DCCP-STD, and also all of the security risks.

The purpose of DCCP-UDP is to allow DCCP to pass through NAT/NAPT devices, and therefore it exposes DCCP to the risks associated with passing through NAT devices. It does not create any new risks with regard to NAT/NAPT devices.

The tunnel encapsulation recommends processing of ICMP messages received for packets sent using DCCP-UDP and translation to allow use by DCCP. [ICMP] describes precautions that are desirable before TCP acts on receipt of ICMP messages. Similar precautions are desirable for endpoints processing ICMP for DCCP-UDP.

DCCP-UDP may also allow DCCP applications to pass through existing firewall devices, if the administrators of the devices so choose. A simple use may either allow all DCCP applications or allow none.

A firewall than interprets this specification could inspect the encapsualted DCCP header to filter based on DCCP information. Full control of DCCP connections by application will require enhancements to firewalls, as discussed in [RFC4340] and related RFCs (e.g. [RFC5595]).

7. IANA Considerations

This document requests IANA to make the allocations described in the following sections.

7.1. UDP Port Allocation

IANA is requested to allocate a UDP port for the dccp-udp service. Use of this port is defined in section Section 3.8

XXX Note: IANA is requested to replace all occurrences of "XXX IANA PORT XXX" by the allocated port value prior to publication. XXX

7.2. DCCP Reset

IANA is requested to assign a new DCCP Reset Code in the DCCP Reset Codes Registry, with the short description "Encapsulated Port Reuse". This code applies to all DCCP congestion control IDs and should be allocated a value less than 120 decimal. Use of this reset code is defined in section Section 3.8

. Section 5.6 of RFC4340 defines three "Data" bytes that are carried by a DCCP Reset. For this Reset Code these are defined as below:

- Data byte 1: The DCCP Packet Type of the DCCP datagram that resulted in the error message.

- Data byte 2 & 3: The encapsulated Source UDP Port from the DCCP-UDP datagram that triggered the , in network order.

XXX Note: IANA is requested to replace all occurrences of "XXX IANA Port Reuse XXX" by the allocated DCCP reset code value prior to publication. XXX

7.3. SDP Attribute Allocation

IANA is requested to allocate the following new SDP attribute ("att-field"):

Contact name: DCCP Working Group

Attribute name: dccp-in-udp

Long-form attribute name in English: DCCP in UDP Encapsulation

Type of attribute: Media level

Subject to charset attribute? No

Purpose of the attribute: See this document section Section 5.1

Allowed attribute values: See this document section Section 5.1

8. Acknowledgments

This document was produced by the DCCP WG. The following contributed during the working group last call:

Andrew Lentvorski, Lloyd Wood, Pasi Sarolahti, Gerrit Renker, Eddie Kohler, Colin Perkins, Dan Wing, Gorry Fairhurst and Tom Phelan.

9. References

9.1. Normative References

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, March 2006.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC5762] Perkins, C., "RTP and the Datagram Congestion Control Protocol (DCCP)", RFC 5762, April 2010.

9.2. Informative References

[ICMP] Gont, ""ICMP attacks against TCP", IETF Work-in-Progress.".

[RFC4787] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, January 2007.

[RFC5238] Phelan, T., "Datagram Transport Layer Security (DTLS) over the Datagram Congestion Control Protocol (DCCP)", RFC 5238, May 2008.

[RFC5595] Fairhurst, G., "The Datagram Congestion Control Protocol (DCCP) Service Codes", RFC 5595, September 2009.

[RFC5596] Fairhurst, G., "Datagram Congestion Control Protocol (DCCP) Simultaneous-Open Technique to Facilitate NAT/ Middlebox Traversal", RFC 5596, September 2009.

[RFC5597] Denis-Courmont, R., "Network Address Translation (NAT) Behavioral Requirements for the Datagram Congestion Control Protocol", BCP 150, RFC 5597, September 2009.

Authors' Addresses

Tom Phelan
Sonus Networks
7 Technology Dr.
Westford, MA 01886
US

Phone: +1 978 614 8456
Email: tphelan@sonusnet.com

Godred Fairhurst
University of Aberdeen
School of Engineering
Fraser Noble Building
Aberdeen, Scotland AB24 3UE
UK

Email: gorry@erg.abdn.ac.uk
URI: http://www.erg.abdn.ac.uk

MMUSIC
Internet-Draft
Intended status: Standards Track
Expires: August 6, 2011

J. Rosenberg
Skype
A. Keranen
Ericsson
B. Lowekamp
Skype
A. Roach
Tekelec

February 2, 2011

TCP Candidates with Interactive Connectivity Establishment (ICE)
draft-ietf-mmusic-ice-tcp-12

Abstract

Interactive Connectivity Establishment (ICE) defines a mechanism for NAT traversal for multimedia communication protocols based on the offer/answer model of session negotiation. ICE works by providing a set of candidate transport addresses for each media stream, which are then validated with peer-to-peer connectivity checks based on Session Traversal Utilities for NAT (STUN). ICE provides a general framework for describing candidates, but only defines UDP-based transport protocols. This specification extends ICE to TCP-based media, including the ability to offer a mix of TCP and UDP-based candidates for a single stream.

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 August 6, 2011.

Copyright Notice

Table of Contents

1. Introduction

Interactive Connectivity Establishment (ICE) [RFC5245] defines a mechanism for NAT traversal for multimedia communication protocols based on the offer/answer model [RFC3264] of session negotiation. ICE works by providing a set of candidate transport addresses for each media stream, which are then validated with peer-to-peer connectivity checks based on Session Traversal Utilities for NAT (STUN) [RFC5389]. However, ICE only defines procedures for UDP-based transport protocols.

There are many reasons why ICE support for TCP is important. Firstly, there are media protocols that only run over TCP. Such protocols are used, for example, for screen sharing and instant messaging [RFC4975]. For these protocols to work in the presence of NAT, unless they define their own NAT traversal mechanisms, ICE support for TCP is needed. In addition, RTP can also run over TCP [RFC4571]. Typically, it is preferable to run RTP over UDP, and not TCP. However, in a variety of network environments, overly restrictive NAT and firewall devices prevent UDP-based communications altogether, but general TCP-based communications are permitted. In such environments, sending RTP over TCP, and thus establishing the media session, may be preferable to having it fail altogether. With this specification, agents can gather UDP and TCP candidates for a media stream, list the UDP ones with higher priority, and then only use the TCP-based ones if the UDP ones fail. This provides a fallback mechanism that allows multimedia communications to be highly reliable.

The usage of RTP over TCP is particularly useful when combined with Traversal Using Relays around NAT (TURN) [RFC5766]. In this case, one of the agents would connect to its TURN server using TCP, and obtain a TCP-based relayed candidate. It would offer this to its peer agent as a candidate. The answerer would initiate a TCP connection towards the TURN server. When that connection is established, media can flow over the connections, through the TURN server. The benefit of this usage is that it only requires the agents to make outbound TCP connections to a server on the public network. This kind of operation is broadly interoperable through NAT and firewall devices. Since it is a goal of ICE and this extension to provide highly reliable communications that "just works" in as a broad a set of network deployments as possible, this use case is particularly important.

This specification extends ICE by defining its usage with TCP candidates. It also defines how ICE can be used with RTP and Secure RTP (SRTP) to provide both TCP and UDP candidates. This specification does so by following the outline of ICE itself, and

calling out the additions and changes necessary in each section of ICE to support TCP candidates.

It should be noted that since TCP NAT traversal is more complicated than with UDP, ICE TCP is not in general as efficient as UDP-based ICE. Discussion about this topic can be found in Appendix A.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

This document uses the same terminology as ICE (see Section 3 of [RFC5245]).

3. Overview of Operation

The usage of ICE with TCP is relatively straightforward. The main area of specification is around how and when connections are opened, and how those connections relate to candidate pairs.

When the agents perform address allocations to gather TCP-based candidates, three types of candidates can be obtained. These are active candidates, passive candidates, and simultaneous-open (S-O) candidates. An active candidate is one for which the agent will attempt to open an outbound connection, but will not receive incoming connection requests. A passive candidate is one for which the agent will receive incoming connection attempts, but not attempt a connection. S-O candidate is one for which the agent will attempt to open a connection simultaneously with its peer.

When gathering candidates from a host interface, the agent typically obtains active, passive, and S-O candidates. Similarly, one can use different techniques for obtaining, e.g., server reflexive, NAT-assisted, tunneled, or relayed candidates of these three types. Connections to servers used for relayed and server reflexive candidates are kept open during ICE processing.

When encoding these candidates into offers and answers, the type of the candidate is signaled. In the case of active candidates, an IP address and port is present, but the port is meaningless, as it is ignored by the peer. As a consequence, active candidates do not need to be physically allocated at the time of address gathering. Rather, the physical allocations, which occur as a consequence of a connection attempt, occur at the time of the connectivity checks.

When the candidates are paired together, active candidates are always paired with passive, and S-O candidates with each other. When a connectivity check is to be made on a candidate pair, each agent determines whether it is to make a connection attempt for this pair.

The actual process of generating connectivity checks, managing the state of the check list, and updating the Valid list, work identically for TCP as they do for UDP.

ICE requires an agent to demultiplex STUN and application layer traffic, since they appear on the same port. This demultiplexing is described in [RFC5245], and is done using the magic cookie and other fields of the message. Stream-oriented transports introduce another wrinkle, since they require a way to frame the connection so that the application and STUN packets can be extracted in order to determine which is which. For this reason, TCP media streams utilizing ICE use the basic framing provided in RFC 4571 [RFC4571], even if the application layer protocol is not RTP.

When TLS or DTLS is used, they are run over the RFC 4571 framing shim, while STUN runs outside of the (D)TLS connection. Pictorially:

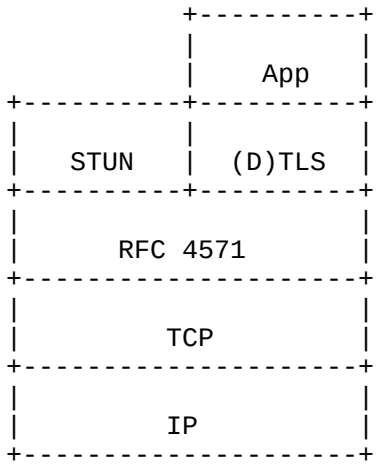


Figure 1: ICE TCP Stack

The implication of this is that, for any media stream protected by (D)TLS, the agent will first run ICE procedures, exchanging STUN messages. Then, once ICE completes, (D)TLS procedures begin. ICE and (D)TLS are thus "peers" in the protocol stack. The STUN messages are not sent over the (D)TLS connection, even ones sent for the purposes of keepalive in the middle of the media session.

When an updated offer is generated by the controlling endpoint, the Session Description Protocol (SDP) extensions for connection oriented media [RFC4145] are used to signal that an existing connection should be used, rather than opening a new one.

4. Sending the Initial Offer

For offerers making use of ICE for TCP streams, the procedures below are used. Main differences compared to UDP candidates are the new methods for gathering candidates, how TCP candidates are prioritized, and how they are encoded in the SDP offer and answer.

4.1. Gathering Candidates

Providers of real-time communications services may decide that it is preferable to have no media at all than it is to have media over TCP. To allow for choice, it is RECOMMENDED that agents be configurable with whether they obtain TCP candidates for real time media.

> Having it be configurable, and then configuring it to be off, is far better than not having the capability at all. An important goal of this specification is to provide a single mechanism that can be used across all types of endpoints. As such, it is preferable to account for provider and network variation through configuration, instead of hard-coded limitations in an implementation. Besides, network characteristics and connectivity assumptions can, and will change over time. Just because an agent is communicating with a server on the public network today, doesn't mean that it won't need to communicate with one behind a NAT tomorrow. Just because an agent is behind a NAT with endpoint-independent mapping today, doesn't mean that tomorrow they won't pick up their agent and take it to a public network access point where there is a NAT with address and port-dependent mapping properties, or one that only allows outbound TCP. The way to handle these cases and build a reliable system is for agents to implement a diverse set of techniques for allocating addresses, so that at least one of them is almost certainly going to work in any situation. Implementors should consider very carefully any assumptions that they make about deployments before electing not to implement one of the mechanisms for address allocation. In particular, implementors should consider whether the elements in the system may be mobile, and connect through different networks with different connectivity. They should also consider whether endpoints which are under their control, in terms of location and network connectivity, would always be under their control. In environments where mobility and user control are possible, a multiplicity of techniques is essential for reliability.

First, agents SHOULD obtain host candidates as described in Section 5.1. Then, each agent SHOULD "obtain" (allocate a placeholder for) an active host candidate for each component of each TCP-capable media stream on each interface that the host has. The agent does not have to yet actually allocate a port for these candidates, but they are used for the creation of the check lists.

Next, the agents SHOULD obtain passive (and possibly S-O) relayed candidates for each component as described in Section 5.5. Each agent SHOULD also allocate a placeholder for an active relayed candidate for each component of each TCP-capable media stream.

The agent SHOULD then obtain server reflexive, NAT-assisted, and/or UDP-tunneled candidates (see Section 5.2, Section 5.3, and Section 5.4). The mechanisms for establishing these candidates and the number of candidates to collect vary from technique to technique. These considerations are discussed in the relevant sections.

It is highly recommended that a host obtains at least one set of host and one set of relayed candidates. Obtaining additional candidates will increase the chance of successfully creating a direct connection.

Once the candidates have been obtained, the agent MUST keep the TCP connections open until ICE processing has completed. See Appendix B for important implementation guidelines.

If a media stream is UDP-based (such as RTP), an agent MAY use an additional host TCP candidate to request a UDP-based candidate from a TURN server (or some other relay with similar functionality). Usage of such UDP candidates follows the procedures defined in ICE for UDP candidates.

Like its UDP counterparts, TCP-based STUN transactions are paced out at one every Ta seconds. This pacing refers strictly to STUN transactions (both Binding and Allocate requests). If performance of the transaction requires establishment of a TCP connection, then the connection gets opened when the transaction is performed.

4.2. Prioritization

The transport protocol itself is a criteria for choosing one candidate over another. If a particular media stream can run over UDP or TCP, the UDP candidates might be preferred over the TCP candidates. This allows ICE to use the lower latency UDP connectivity if it exists, but fallback to TCP if UDP doesn’t work.

In Section 4.1.2.1. of [RFC5245] a recommended formula for UDP ICE candidate prioritization is defined. For the TCP candidates the same formula and candidate type preferences SHOULD be used and the RECOMMENDED type preferences for the new candidate types defined in this document (see Section 5) are 105 for NAT-assisted candidates and 75 for UDP-tunneled candidates.

When both UDP and TCP candidates are offered for the same media stream, and one transport protocol should be preferred over the other, the type preferences for the preferred transport protocol candidates SHOULD be increased and/or the type preferences for the other transport protocol candidates SHOULD be decreased. How much the values should be increased or decreased depends on whether it is more important to choose certain transport protocol or certain candidate type. If the candidate type is more important (e.g., even if UDP is preferred, TCP host candidates are preferred over UDP server reflexive candidates) changing type preference values by one for the other transport protocol candidates is enough. On the other hand, if the transport protocol is more important (e.g., any UDP candidate is preferred over any TCP candidate) all the preferred transport protocol candidates SHOULD have type preference higher than the other transport protocol candidates. However, it is RECOMMENDED that the relayed candidates are still preferred lower than the other candidate types. For RTP-based media streams, it is RECOMMENDED that UDP candidates are preferred over TCP candidates.

With TCP candidates the local-preference part of the recommended priority formula is updated to include also the directionality (active, passive, or simultaneous-open) of the TCP connection. The RECOMMENDED local-preference is then defined as:

```
local-preference = (2^13) * direction-pref + other-pref
```

The direction-pref MUST be between 0 and 7, with 7 being the most preferred. The other-pref MUST be between 0 and 8191, with 8191 being the most preferred. It is RECOMMENDED that the host, UDP-tunneled, and relayed TCP candidates have the direction-pref assigned as follows: 6 for active, 4 for passive, and 2 for S-O. For the NAT-assisted and server reflexive candidates the RECOMMENDED values are: 6 for S-O, 4 for active, and 2 for passive.

> The preference priorities listed here are simply recommendations that try to strike a balance between success probability and resulting path's efficiency. Depending on the scenario where ICE TCP is used, different values may be appropriate. For example, if the overhead of a UDP tunnel is not an issue, those candidates should be prioritized higher since they are likely to have a high success probability. Also, simultaneous-open is prioritized

higher than active and passive candidates for NAT-assisted and server reflexive candidates since if TCP S-O is supported by the operating systems of both endpoints, it should work at least as well as the act-pass approach. If an implementation is uncertain whether S-O candidates are supported, it may be reasonable to prioritize them lower. For host, UDP-tunneled, and relayed candidates the S-O candidates are prioritized lower than active and passive since act-pass candidates should work with them at least as well as the S-O candidates.

If any two candidates have the same type-preference and direction-pref, they MUST have a unique other-pref. With this specification, this usually only happens with multi-homed hosts, in which case other-pref is the preference for the particular IP address from which the candidate was obtained. When there is only a single IP address, this value SHOULD be set to the maximum allowed value (8191).

4.3. Choosing Default Candidates

The default candidate is chosen primarily based on the likelihood of it working with a non-ICE peer. When media streams supporting mixed modes (both TCP and UDP) are used with ICE, it is RECOMMENDED that, for real-time streams (such as RTP), the default candidates be UDP-based. However, the default SHOULD NOT be a simultaneous-open candidate.

If a media stream is inherently TCP-based, the agent MUST select an active candidate as the default. This ensures proper directionality of connection establishment for NAT traversal with non-ICE implementations.

4.4. Lite Implementation Requirements

If an offerer meets the criteria for the lite mode as described in Appendix A of [RFC5245] (i.e., it will always have a public, globally unique IP address), it MAY use the lite mode of ICE also for TCP candidates. In the lite mode, for the TCP candidates, only passive host candidates are gathered, but the offerer gathers one additional active host candidate to be the default candidate. Otherwise the procedures described for lite mode in [RFC5245] apply also to TCP candidates. If UDP and TCP candidates are mixed in a media stream, the mode (lite or full) applies to both UDP and TCP candidates.

4.5. Encoding the SDP

TCP-based candidates are encoded into a=candidate lines like the UDP candidates described in [RFC5245]. However, the transport protocol (i.e., value of the transport-extension token defined in [RFC5245]

Section 15.1) is set to "TCP" and the connection type (active, passive, or S-O) is encoded using a new extension attribute. With TCP candidates, the candidate-attribute syntax with Augmented BNF [RFC5234] is then:

```
candidate-attribute   = "candidate" ":" foundation SP component-id SP
                        "TCP" SP
                        priority SP
                        connection-address SP
                        port SP
                        cand-type
                        [SP rel-addr]
                        [SP rel-port]
                        SP tcp-type-ext
                        *(SP extension-att-name SP
                             extension-att-value)

tcp-type-ext          = "tcptype" SP tcp-type
tcp-type              = "act" / "pass" / "so"
```

The connection-address and port encoded into the candidate attribute for active candidates MUST be set to the IP address that will be used for the attempt, but the port(s) MUST be set to 9 (i.e., Discard). For active relayed candidates, the value for connection-address MUST be identical to the IP address of a passive or simultaneous-open candidate from the same relay server.

If the default candidate is TCP-based, the agent MUST include the a=setup and a=connection attributes from RFC 4145 [RFC4145], following the procedures defined there as if ICE was not in use. In particular, if an agent is the answerer, the a=setup attribute MUST meet the constraints in RFC 4145 based on the value in the offer. Since an ICE ICE offerer always uses an active candidate as default, an ICE ICE answerer will always use a passive candidate as default and include the a=setup:passive attribute in the answer.

If an agent is utilizing SRTP [RFC3711], it MAY include a mix of UDP and TCP candidates. If ICE selects a TCP candidate pair, the agent MUST still utilize SRTP, but run it over the connection established by ICE. The alternative, RTP over TLS, MUST NOT be used. This allows for the higher layer protocols (the security handshakes and media transport) to be independent of the underlying transport protocol. In the case of DTLS-SRTP [RFC5764], the directionality attributes (a=setup) are utilized strictly to determine the direction of the DTLS handshake. Directionality of the TCP connection establishment are determined by the ICE attributes and procedures defined here.

If an agent is securing non-RTP media over TCP/TLS, the SDP MUST be constructed as described in RFC 4572 [RFC4572]. The directionality attributes (a=setup) are utilized strictly to determine the direction of the TLS handshake. Directionality of the TCP connection establishment are determined by the ICE attributes and procedures defined here.

Examples of SDP offers and answers with ICE TCP extensions are shown in Appendix C.

5. Candidate Collection Techniques

The following sections discuss a number of techniques that can be used to obtain candidates for use with ICE TCP. It is important to note that this list is not intended to be exhaustive, nor is implementation of any specific technique beyond host candidates (Section 5.1) considered mandatory.

Implementors are encouraged to implement as many of the following techniques from the following list as is practical, as well as to explore additional NAT-traversal techniques beyond those discussed in this document. However, to get a reasonable success ratio, one SHOULD implement at least one relayed technique (e.g., TURN) and one technique for discovering the address given for the host by a NAT (e.g., STUN).

To increase the success probability with the techniques described below and to aid with transition to IPv6, implementors SHOULD take particular care to include both IPv4 and IPv6 candidates as part of the process of gathering candidates. If the local network or host does not support IPv6 addressing, then clients SHOULD make use of other techniques, e.g., Teredo [RFC4380] or SOCKS IPv4-IPv6 gatewaying [RFC3089], for obtaining IPv6 candidates.

While implementations SHOULD support as many techniques as feasible, they SHOULD also consider which of them to use if multiple options are available. Since different candidates are paired with each other, offering a large amount of candidates results in a large checklist and potentially long lasting connectivity checks. For example, using multiple NAT-assisted techniques with the same NAT usually results only in redundant candidates and similarly out of multiple different UDP tunneling or relaying techniques with similar features using just one is often enough.

5.1. Host Candidates

Host candidates are the most simple candidates since they only require opening TCP sockets on the host's interfaces and sending/ receiving connectivity checks from them. However, if the hosts are behind different NATs, host candidates usually fail to work. On the other hand, if there are no NATs between the hosts, host candidates are the most efficient method since they require no additional NAT traversal protocols or techniques.

For each TCP-capable media stream the agent wishes to use (including ones, like RTP, which can either be UDP or TCP), the agent SHOULD obtain two host candidates (each on a different port) for each component of the media stream on each interface that the host has - one for the simultaneous-open, and one for the passive candidate. If an agent is not capable of acting in one of these modes it would omit those candidates.

5.2. Server Reflexive Candidates

Server reflexive techniques aim to discover the address a NAT has given for the host by asking that from a server on the other side of the NAT and then creating proper bindings (unless such already exist) on the NATs with connectivity checks sent between the hosts. Success of these techniques depends on the NATs' mapping and filtering behavior [RFC5382] and also whether the NATs and hosts support the TCP simultaneous-open technique.

A widely used protocol for obtaining server reflexive candidates is STUN, whose TCP-specific behavior is described in [RFC5389] Section 7.2.2.

5.3. NAT-Assisted Candidates

NAT-assisted techniques communicate with the NATs directly and this way discover the address NAT has given to the host and also create proper bindings on the NATs. The benefit of these techniques over the server reflexive techniques is that the NATs can adjust their mapping and filtering behavior so that connections can be successfully created. A downside of NAT-assisted techniques is that they commonly allow communicating only with a NAT that is in the same subnet as the host and thus often fail in scenarios with multiple layers of NATs. These techniques also rely on NATs supporting the specific protocols and that the NATs allow the users to modify their behavior.

These candidates are encoded in the ICE offer and answer like the server reflexive candidates but they (commonly) use a higher priority

(as described in Section 4.2) and hence are tested before the server reflexive candidates.

Currently, the UPnP forum's Internet Gateway Device (IGD) protocol [UPnP-IGD] and the NAT Port Mapping Protocol (PMP) [I-D.cheshire-nat-pmp] are widely supported NAT-assisted techniques. Other known protocols include SOCKS [RFC1928], Realm Specific IP (RSIP) [RFC3103], and SIMCO [RFC4540]. Also, MIDCOM MIB [RFC5190] defines an SNMP-based mechanism for controlling NATs.

5.4. UDP-Tunneled Candidates

UDP-tunneled NAT traversal techniques utilize the fact that UDP NAT traversal is simpler and more efficient than TCP NAT traversal. With these techniques, the TCP packets (or possibly complete IP packets) are encapsulated in UDP packets. Because of the encapsulation these techniques increase the overhead for the connection and may require support from both of the endpoints, but on the other hand UDP tunneling commonly results in reliable and fairly simple TCP NAT traversal.

UDP-tunneled candidates can be encoded in the ICE offer and answer either as relayed or server reflexive candidates, depending on whether the tunneling protocol utilizes a relay between the hosts.

For example, the Teredo protocol [RFC4380] [RFC6081] provides automatic UDP tunneling and IPv6 interworking. The Teredo UDP tunnel is visible to the host application as an IPv6 address and thus Teredo candidates are encoded as IPv6 addresses.

5.5. Relayed Candidates

Relaying packets through a relay server is often the NAT traversal technique that has the highest success probability: communicating via a relay that is in the public Internet looks like normal client-server communication for the NATs and that is supported in practice by all existing NATs, regardless of their filtering and mapping behavior. However, using a relay has several drawbacks, e.g., it usually results in a sub-optimal path for the packets, the relay needs to exist and it needs to be discovered, the relay is a possible single point of failure, relaying consumes potentially a lot of resources of the relay server, etc. Therefore, relaying is often used as the last resort when no direct path can be created with other NAT traversal techniques.

With relayed candidates the host commonly needs to obtain only a passive candidate since any of the peer's server reflexive (and NAT-assisted if the peer can communicate with the outermost NAT) active

candidates should work with the passive relayed candidate. However, if the relay is behind a NAT or a firewall, using also active and S-O candidates will increase success probability.

Relaying protocols capable of relaying TCP connections include TURN TCP [RFC6062] and SOCKS [RFC1928] (which can also be used for IPv4-IPv6 gatewaying [RFC3089]). It is also possible to use, e.g., an SSH [RFC4250] tunnel as a relayed candidate if a suitable server is available and the server permits this.

6. Receiving the Initial Offer and Answer

Handling an ICE offer with TCP candidates works in a similar way as with UDP candidates. First, ICE support is verified (including the check for ice-mismatch described in Section 5.1 of [RFC5245]) and agent roles are determined. Candidates are gathered using the techniques described in Section 5 and prioritized as described in Section 4.2. Default candidates are selected taking into account considerations of Section 4.3. The SDP answer is encoded as in Section 4.3 of [RFC5245] with the exception of TCP candidates whose encoding was described in Section 4.5.

When the offerer receives the initial answer, it also verifies ICE support and determines its role. If both of the agents use lite implementations, the offerer takes the controlling role and uses the procedures defined in [RFC4145] to select the most preferred candidate pair with a new offer.

6.1. Considerations with Two Lite Agents

If both agents are using the lite mode, and if the offerer uses a=setup:active attribute [RFC4145] in the new offer, the offerer MAY initiate the TCP connection on the selected pair in parallel with the new offer to speedup the connection establishment. Consequently, the answerer MUST still accept incoming TCP connections to any of the passive candidates it listed in the answer, from any of the IP addresses the offerer listed in the initial offer.

If the answerer receives the new offer matching to the candidate pair where connection was already created in parallel with the new offer, it MUST accept the offer and respond to it while keeping the already created connection. If the connection that was created in parallel with the new offer does not match to the candidate pair in the new offer, the connection MUST be closed and ICE restart SHOULD be performed.

Since the connection endpoints are not authenticated using the connectivity checks in the scenario where both agents use the lite mode, unless media-level security (e.g., TLS) is used, it is RECOMMENDED to use the full mode instead. For more lite vs. full implementation considerations, see Appendix A of [RFC5245].

6.2. Forming the Check Lists

As with UDP, checklists are formed only by full ICE implementations. When forming candidate pairs, the following types of TCP candidates can be paired with each other:

Local Candidate	Remote Candidate
tcp-so	tcp-so
tcp-act	tcp-pass
tcp-pass	tcp-act

When the agent prunes the check list, it MUST also remove any pair for which the local candidate is a passive TCP candidate. With pruning, the NAT-assisted candidates are treated like server reflexive candidates if the base is used also as a host candidate.

The remainder of check list processing works like in the UDP case.

7. Connectivity Checks

The TCP connectivity checks, like with UDP, are generated only by full implementations. The TCP candidate pairs are in the same checklist with the UDP candidate pairs and they are scheduled for connectivity checks based on the priority order.

7.1. STUN Client Procedures

When an agent wants to send a TCP-based connectivity check, it first opens a TCP connection, if none yet exists, for the 5-tuple defined by the candidate pair for which the check is to be sent. This connection is opened from the local candidate of the pair to the remote candidate of the pair. If the local candidate is tcp-act, the agent MUST open a connection from the interface associated with that local candidate. This connection SHOULD be opened from an unallocated port. For host candidates, this is readily done by connecting from the local candidate’s interface. For relayed, NAT-assisted, and UDP-tunneled candidates, the agent may need to use additional procedures specific to the protocol.

Once the connection is established, the agent MUST utilize the shim defined in RFC 4571 [RFC4571] for the duration this connection remains open. The STUN Binding requests and responses are sent on top of this shim, so that the length field defined in RFC 4571 precedes each STUN message. If TLS or DTLS-SRTP is to be utilized for the media session, the TLS or DTLS-SRTP handshakes will take place on top of this shim as well. However, they only start once ICE processing has completed. In essence, the TLS or DTLS-SRTP handshakes are considered a part of the media protocol. STUN is never run within the TLS or DTLS-SRTP session.

If the TCP connection cannot be established, the check is considered to have failed, and a full-mode agent MUST update the pair state to Failed in the check list.

Once the connection is established, client procedures are identical to those for UDP candidates. However, retransmissions of the STUN connectivity check messages are not needed, since TCP takes care of reliable delivery of the messages. Note also that STUN responses received on an active TCP candidate will typically produce a remote peer reflexive candidate.

7.2. STUN Server Procedures

An ICE TCP agent, full or lite, MUST be prepared to receive incoming TCP connection requests on the base of any TCP candidate that is simultaneous-open or passive. When the connection request is received, the agent MUST accept it. The agent MUST utilize the framing defined in RFC 4571 [RFC4571] for the lifetime of this connection. Due to this framing, the agent will receive data in discrete frames. Each frame could be media (such as RTP or SRTP), TLS, DTLS, or STUN packets. The STUN packets are extracted as described in Section 10.2.

Once the connection is established, STUN server procedures are identical to those for UDP candidates. Note that STUN requests received on a passive TCP candidate will typically produce a remote peer reflexive candidate.

8. Concluding ICE Processing

If there are TCP candidates for a media stream, a controlling agent MUST use the regular selection algorithm.

When ICE processing for a media stream completes, each agent SHOULD close all TCP connections (that were opened due to this ICE session) except the ones between the candidate pairs selected by ICE.

These two rules are related; the closure of connection on completion of ICE implies that a regular selection algorithm has to be used. This is because aggressive selection might cause transient pairs to be selected. Once such a pair was selected, the agents would close the other connections, one of which may be about to be selected as a better choice. This race condition may result in TCP connections being accidentally closed for the pair that ICE selects.

9. Subsequent Offer/Answer Exchanges

9.1. ICE Restarts

If an ICE restart occurs for a media stream with TCP candidate pairs that have been selected by ICE, the agents MUST NOT close the connections after the restart. In the offer or answer that causes the restart, an agent MAY include a simultaneous-open candidate whose transport address matches the previously selected candidate. If both agents do this, the result will be a simultaneous-open candidate pair matching an existing TCP connection. In this case, the agents MUST NOT attempt to open a new connection (or start new TLS or DTLS-SRTP procedures). Instead, that existing connection is reused and STUN checks are performed.

Once the restart completes, if the selected pair does not match the previously selected pair, the TCP connection for the previously selected pair SHOULD be closed by the agent.

10. Media Handling

10.1. Sending Media

When sending media, if the selected candidate pair matches an existing TCP connection, that connection MUST be used for sending media.

The framing defined in RFC 4571 MUST be used when sending media. For media streams that are not RTP-based and do not normally use RFC 4571, the agent treats the media stream as a byte stream, and assumes that it has its own framing of some sort. It then takes an arbitrary number of bytes from the byte stream, and places that as a payload in the RFC 4571 frames, including the length. Next, the sender checks to see if the resulting set of bytes would be viewed as a STUN packet based on the rules in Sections 6 and 8 of [RFC5389]. This includes a check on the most significant two bits, the magic cookie, the length, and the fingerprint. If, based on those rules, the bytes would be

viewed as a STUN message, the sender SHOULD utilize a different number of bytes so that the length checks will fail. Though it is normally highly unlikely that an arbitrary number of bytes from a byte stream would resemble a STUN packet based on all of the checks, it can happen if the content of the application stream happens to contain a STUN message (for example, a file transfer of logs from a client which includes STUN messages).

If TLS or DTLS-SRTP procedures are being utilized to protect the media stream, those procedures start at the point that media is permitted to flow, as defined in the ICE specification [RFC5245]. The TLS or DTLS-SRTP handshakes occur on top of the RFC 4571 shim, and are considered part of the media stream for purposes of this specification.

10.2. Receiving Media

The framing defined in RFC 4571 MUST be used when receiving media. For media streams that are not RTP-based and do not normally use RFC 4571, the agent extracts the payload of each RFC 4571 frame, and determines if it is a STUN or an application layer data based on the procedures in ICE [RFC5245]. If media is being protected with DTLS-SRTP, the DTLS, RTP and STUN packets are demultiplexed as described in Section 5.1.2 [RFC5764].

For non-STUN data, the agent appends this to the ongoing byte stream collected from the frames. It then parses the byte stream as if it had been directly received over the TCP connection. This allows for ICE TCP to work without regard to the framing mechanism used by the application layer protocol.

11. Connection Management

11.1. Connections Formed During Connectivity Checks

Once a TCP or TCP/TLS connection is opened by ICE for the purpose of connectivity checks, its life cycle depends on how it is used. If that candidate pair is selected by ICE for usage for media, an agent SHOULD keep the connection open until:

- The session terminates
- The media stream is removed
- An ICE restart takes place, resulting in the selection of a different candidate pair.

In these cases, the agent SHOULD close the connection when that event occurs. This applies to both agents in a session, in which case usually one of the agents will end up closing the connection first.

If a connection has been selected by ICE, an agent MAY close it anyway. As described in the next paragraph, this will cause it to be reopened almost immediately, and in the interim media cannot be sent. Consequently, such closures have a negative effect and are NOT RECOMMENDED. However, there may be cases where an agent needs to close a connection for some reason.

If an agent needs to send media on the selected candidate pair, and its TCP connection has closed, either on purpose or due to some error, then:

- If the agent's local candidate is tcp-act or tcp-so, it MUST reopen a connection to the remote candidate of the selected pair.

- If the agent's local candidate is tcp-pass, the agent MUST await an incoming connection request, and consequently, will not be able to send media until it has been opened.

If the TCP connection is established, the framing of RFC 4571 is utilized. If the agent opened the connection, it MUST send a STUN connectivity check. An agent MUST be prepared to receive a connectivity check over a connection it opened or accepted (note that this is true in general; ICE requires that an agent be prepared to receive a connectivity check at any time, even after ICE processing completes). If an agent receives a connectivity check after re-establishment of the connection, it MUST generate a triggered check over that connection in response if it has not already sent a check. Once an agent has sent a check and received a successful response, the connection is considered Valid and media can be sent (which includes a TLS or DTLS-SRTP session resumption or restart).

If the TCP connection cannot be established, the controlling agent SHOULD restart ICE for this media stream. This will happen in cases where one of the agents is behind a NAT with connection-dependent mapping properties [RFC5382].

11.2. Connections Formed for Gathering Candidates

If the agent opened a connection to a STUN server, or another similar server, for the purposes of gathering a server reflexive candidate, that connection SHOULD be closed by the client once ICE processing has completed. This happens irregardless of whether the candidate learned from the server was selected by ICE.

If the agent opened a connection to a TURN server for the purposes of gathering a relayed candidate, that connection MUST be kept open by the client for the duration of the media session if a relayed candidate from the TURN server was selected by ICE. Otherwise, the connection to the TURN server SHOULD be closed once ICE processing completes.

If, despite efforts of the client, a TCP connection to a TURN server fails during the lifetime of the media session utilizing a transport address allocated by that server, the client SHOULD reconnect to the TURN server, obtain a new allocation, and restart ICE for that media stream. Similar measures SHOULD apply also to other type of relaying servers.

12. Security Considerations

The main threat in ICE is hijacking of connections for the purposes of directing media streams to DoS targets or to malicious users. When full implementations are used, ICE TCP prevents that by only using TCP connections that have been validated. Validation requires a STUN transaction to take place over the connection. This transaction cannot complete without both participants knowing a shared secret exchanged in the rendezvous protocol used with ICE, such as SIP [RFC3261]. This shared secret, in turn, is protected by that protocol exchange. In the case of SIP, the usage of the sips mechanism is RECOMMENDED. When this is done, an attacker, even if it knows or can guess the port on which an agent is listening for incoming TCP connections, will not be able to open a connection and send media to the agent.

If both agents use the lite mode, no connectivity checks are sent, and additional procedures (e.g., media-level security) are needed to validate the connection. The lack of connectivity checks is especially problematic if one of the hosts is behind a NAT and has an address from a private address space: the peer may accidentally connect to a host in a different subnet that uses the same private address space. This is one of the reasons why the lite mode is not appropriate for an ICE agent located behind a NAT.

A more detailed analysis of this attack and the various ways ICE prevents it are described in [RFC5245]. Those considerations apply to this specification.

13. IANA Considerations

There are no IANA considerations associated with this specification.

14. Acknowledgements

The authors would like to thank Tim Moore, Saikat Guha, Francois Audet, Roni Even, Simon Perreault, Alfred Heggestad, and Hadriel Kaplan for the reviews and input on this document. Special thanks to Marc Petit-Huguenin for providing the SDP examples.

15. References

15.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[RFC4145] Yon, D. and G. Camarillo, "TCP-Based Media Transport in the Session Description Protocol (SDP)", RFC 4145, September 2005.

[RFC4571] Lazzaro, J., "Framing Real-time Transport Protocol (RTP) and RTP Control Protocol (RTCP) Packets over Connection-Oriented Transport", RFC 4571, July 2006.

[RFC4572] Lennox, J., "Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP)", RFC 4572, July 2006.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC5764] McGrew, D. and E. Rescorla, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP)", RFC 5764, May 2010.

15.2. Informative References

[RFC1928] Leech, M., Ganis, M., Lee, Y., Kuris, R., Koblas, D., and L. Jones, "SOCKS Protocol Version 5", RFC 1928, March 1996.

[RFC3089] Kitamura, H., "A SOCKS-based IPv6/IPv4 Gateway Mechanism", RFC 3089, April 2001.

[RFC3103] Borella, M., Grabelsky, D., Lo, J., and K. Taniguchi, "Realm Specific IP: Protocol Specification", RFC 3103, October 2001.

[RFC3261] 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.

[RFC4250] Lehtinen, S. and C. Lonvick, "The Secure Shell (SSH) Protocol Assigned Numbers", RFC 4250, January 2006.

[RFC4380] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, February 2006.

[RFC4540] Stiemerling, M., Quittek, J., and C. Cadar, "NEC's Simple Middlebox Configuration (SIMCO) Protocol Version 3.0", RFC 4540, May 2006.

[RFC4975] Campbell, B., Mahy, R., and C. Jennings, "The Message Session Relay Protocol (MSRP)", RFC 4975, September 2007.

[RFC5190] Quittek, J., Stiemerling, M., and P. Srisuresh, "Definitions of Managed Objects for Middlebox Communication", RFC 5190, March 2008.

[RFC5382] Guha, S., Biswas, K., Ford, B., Sivakumar, S., and P. Srisuresh, "NAT Behavioral Requirements for TCP", BCP 142, RFC 5382, October 2008.

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.

[RFC5766] Mahy, R., Matthews, P., and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", RFC 5766, April 2010.

[RFC6062] Perreault, S. and J. Rosenberg, "Traversal Using Relays around NAT (TURN) Extensions for TCP Allocations", RFC 6062, November 2010.

[RFC6081] Thaler, D., "Teredo Extensions", RFC 6081, January 2011.

[I-D.cheshire-nat-pmp] Cheshire, S., "NAT Port Mapping Protocol (NAT-PMP)", draft-cheshire-nat-pmp-03 (work in progress), April 2008.

[UPnP-IGD] Warrier, U., Iyer, P., Pennerath, F., Marynissen, G., Schmitz, M., Siddiqi, W., and M. Blaszczak, "Internet Gateway Device (IGD) Standardized Device Control Protocol V 1.0", November 2001.

[IMC05] Guha, S. and P. Francis, "Characterization and Measurement of TCP Traversal through NATs and Firewalls", Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement, 2005.

Appendix A. Limitations of ICE TCP

Compared to UDP-based ICE, ICE TCP has in general lower success probability for enabling connectivity without a relay if both of the hosts are behind a NAT. This happens because many of the currently deployed NATs have endpoint-dependent mapping behavior or they do not support the flow of TCP hand shake packets seen in case of TCP simultaneous-open: e.g., some NATs do not allow incoming TCP SYN packets from an address where a SYN packet has been sent to recently or the subsequent SYNACK is not processed properly.

It has been reported in [IMC05] that with the population of NATs deployed at the time of the measurements (2005), one of the NAT traversal techniques described here, TCP simultaneous-open, worked in roughly 45% of the cases. Also, all operating systems do not implement TCP simultaneous-open properly and thus are not able to use such candidates. However, when more NATs comply with the requirements set by [RFC5382] and operating system TCP stacks are fixed, the success probability of simultaneous-open is likely to increase. Also, it is important to implement additional techniques with higher success ratio, such as Teredo, whose success in different scenarios is described in Figure 1 of [RFC6081].

Finally, it should be noted that implementing various techniques listed in Section 5 should increase the success probability, but many of these techniques require support from the endpoints and/or from

some network elements (e.g., from the NATs). Without comprehensive experimental data on how well different techniques are supported the actual increase of success probability is hard to evaluate.

Appendix B. Implementation Considerations for BSD Sockets

This specification requires unusual handling of TCP connections, the implementation of which in traditional BSD socket APIs is non-trivial.

In particular, ICE requires an agent to obtain a local TCP candidate, bound to a local IP and port, and then from that local port, initiate a TCP connection (e.g., to the STUN server, in order to obtain server reflexive candidates, to the TURN server, to obtain a relayed candidate, or to the peer as part of a connectivity check), and be prepared to receive incoming TCP connections (for passive and simultaneous-open candidates). A "typical" BSD socket is used either for initiating or receiving connections, and not for both. The code required to allow incoming and outgoing connections on the same local IP and port is non-obvious. The following pseudocode, contributed by Saikat Guha, has been found to work on many platforms:

```
for i in 0 to MAX
   sock_i = socket()
   set(sock_i, SO_REUSEADDR)
   bind(sock_i, local)

listen(sock_0)
connect(sock_1, stun)
connect(sock_2, remote_a)
connect(sock_3, remote_b)
```

The key here is that, prior to the listen() call, the full set of sockets that need to be utilized for outgoing connections must be allocated and bound to the local IP address and port. This number, MAX, represents the maximum number of TCP connections to different destinations that might need to be established from the same local candidate. This number can be potentially large for simultaneous-open candidates. If a request forks, ICE procedures may take place with multiple peers. Furthermore, for each peer, connections would need to be established to each passive or simultaneous-open candidate for the same component. If we assume a worst case of 5 forked branches, and for each peer, five simultaneous-open candidates, that results in MAX=25.

Appendix C. SDP Examples

This section shows two examples of SDP offer and answer when the ICE TCP extension is used. Both examples are based on the simplified topology of Figure 8 in [RFC5245], with the same IP addresses. The examples shown here should be considered as strictly informative.

In the first example, the offer contains only TCP candidates (lines folded in examples to satisfy RFC formatting rules):

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 10.0.1.1
s=
c=IN IP4 192.0.2.3
t=0 0
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
m=audio 45664 TCP/RTP/AVP 0
b=RS:0
b=RR:0
a=rtpmap:0 PCMU/8000
a=setup:active
a=connection:new
a=candidate:1 1 TCP 2128609279 10.0.1.1 9 typ host tcptype act
a=candidate:2 1 TCP 2124414975 10.0.1.1 8998 typ host tcptype pass
a=candidate:3 1 TCP 2120220671 10.0.1.1 8999 typ host tcptype so
a=candidate:4 1 TCP 1688207359 192.0.2.3 9 typ srflx raddr 10.0.1.1
  rport 9 tcptype act
a=candidate:5 1 TCP 1684013055 192.0.2.3 45664 typ srflx raddr
  10.0.1.1 rport 8998 tcptype pass
a=candidate:6 1 TCP 1692401663 192.0.2.3 45687 typ srflx raddr
  10.0.1.1 rport 8999 tcptype so
```

The answer to that offer could look like this:

```
v=0
o=bob 2808844564 2808844564 IN IP4 192.0.2.1
s=
c=IN IP4 192.0.2.1
t=0 0
a=ice-pwd:YH75Fviy6338Vbrhrlp8Yh
a=ice-ufrag:9uB6
m=audio 3478 TCP/RTP/AVP 0
b=RS:0
b=RR:0
a=setup:passive
a=connection:new
a=rtpmap:0 PCMU/8000
a=candidate:1 1 TCP 2128609279 192.0.2.1 9 typ host tcptype act
a=candidate:2 1 TCP 2124414975 192.0.2.1 3478 typ host tcptype pass
a=candidate:3 1 TCP 2120220671 192.0.2.1 3482 typ host tcptype so
```

In the second example, UDP and TCP media streams are mixed but S-O candidates are omitted due to hosts not supporting TCP simultaneous-open and UDP candidates are preferred (but preference order for candidate types is kept the same) by decreasing the TCP candidate type preferences by one (i.e., using type preference 125 for the host candidates and 99 for the server reflexive candidates):

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 10.0.1.1
s=
c=IN IP4 192.0.2.3
t=0 0
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
m=audio 45664 RTP/AVP 0
b=RS:0
b=RR:0
a=rtpmap:0 PCMU/8000
a=candidate:1 1 TCP 2111832063 10.0.1.1 9 typ host tcptype act
a=candidate:2 1 TCP 2107637759 10.0.1.1 9012 typ host tcptype pass
a=candidate:3 1 TCP 1671430143 192.0.2.3 9 typ srflx raddr 10.0.1.1
  rport 9 tcptype act
a=candidate:4 1 TCP 1667235839 192.0.2.3 44642 typ srflx raddr
  10.0.1.1 rport 9012 tcptype pass
a=candidate:5 1 UDP 2130706431 10.0.1.1 8998 typ host
a=candidate:6 1 UDP 1694498815 192.0.2.3 45664 typ srflx raddr
  10.0.1.1 rport 8998
```

The corresponding answer could look like this:

```
v=0
o=bob 2808844564 2808844564 IN IP4 192.0.2.1
s=
c=IN IP4 192.0.2.1
t=0 0
a=ice-pwd:YH75Fviy6338Vbrhrlp8Yh
a=ice-ufrag:9uB6
m=audio 3478 RTP/AVP 0
b=RS:0
b=RR:0
a=rtpmap:0 PCMU/8000
a=candidate:1 1 TCP 2111832063 192.0.2.1 9 typ host tcptype act
a=candidate:2 1 TCP 2107637759 192.0.2.1 3478 typ host tcptype pass
a=candidate:3 1 UDP 2130706431 192.0.2.1 3478 typ host
```

Authors' Addresses

Jonathan Rosenberg
Skype

Email: jdrosen@jdrosen.net
URI: http://www.jdrosen.net

Ari Keranen
Ericsson
Hirsalantie 11
02420 Jorvas
Finland

Email: ari.keranen@ericsson.com

Bruce B. Lowekamp
Skype

Email: bbl@lowekamp.net

Adam Roach
Tekelec
17210 Campbell Rd.
Suite 250
Dallas, TX 75252
US

Email: adam@nostrum.com

MMUSIC WG
Internet-Draft
Intended status: Standards Track
Expires: August 22, 2011

M. Garcia-Martin
Ericsson
S. Veikkolainen
Nokia

February 18, 2011

Session Description Protocol (SDP) Extension For Setting Up Audio and Video Media Streams Over Circuit-Switched Bearers In The Public Switched Telephone Network (PSTN) draft-ietf-mmusic-sdp-cs-06

Abstract

This memo describes use cases, requirements, and protocol extensions for using the Session Description Protocol (SDP) Offer/Answer model for establishing audio and video media streams over circuit-switched bearers in the Public Switched Telephone Network (PSTN).

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 August 22, 2011.

Copyright Notice

Table of Contents

1. Introduction

The Session Description Protocol (SDP) [RFC4566] is intended for describing multimedia sessions for the purposes of session announcement, session invitation, and other forms of multimedia session initiation. SDP is most commonly used for describing media streams that are transported over the Real-Time Transport Protocol (RTP) [RFC3550], using the profiles for audio and video media defined in RTP Profile for Audio and Video Conferences with Minimal Control [RFC3551].

However, SDP can be used to describe other transport protocols than RTP. Previous work includes SDP conventions for describing ATM bearer connections [RFC3108] and the Message Session Relay Protocol [RFC4975].

SDP is commonly carried in Session Initiation Protocol (SIP) [RFC3261] messages in order to agree on a common media description among the endpoints. An Offer/Answer Model with Session Description Protocol (SDP) [RFC3264] defines a framework by which two endpoints can exchange SDP media descriptions and come to an agreement as to which media streams should be used, along with the media related parameters.

In some scenarios it might be desirable to establish the media stream over a circuit-switched bearer connection even if the signaling for the session is carried over an IP bearer. An example of such a scenario is illustrated with two mobile devices capable of both circuit-switched and packet-switched communication over a low-bandwidth radio bearer. The radio bearer may not be suitable for carrying real-time audio or video media, and using a circuit-switched bearer would offer a better perceived quality of service. So, according to this scenario, SDP and its higher layer session control protocol (e.g., the Session Initiation Protocol (SIP) [RFC3261]) are used over regular IP connectivity, while the audio or video is received through the classical circuit-switched bearer.

Setting up a signaling relationship in the IP domain instead of just setting up a circuit-switched call offers also the possibility of negotiating in the same session other IP based media that is not sensitive to jitter and delay, for example, text messaging or presence information.

At a later point in time the mobile device might move to an area where a high-bandwidth packet-switched bearer, for example a Wireless Local Area Network (WLAN) connection, is available. At this point the mobile device may perform a handover and move the audio or video media streams over to the high-speed bearer. This implies a new

exchange of SDP Offer/Answer that lead to a re-negotiation of the media streams.

Other use cases exist. For example, and endpoint might have at its disposal circuit-switched and packet-switched connectivity, but the same audio or video codecs are not feasible for both access networks. For example, the circuit-switched audio or video stream supports narrow-bandwidth codecs, while the packet-switched access allows any other audio or video codec implemented in the endpoint. In this case, it might be beneficial for the endpoint to describe different codecs for each access type and get an agreement on the bearer together with the remote endpoint.

There are additional use cases related to third party call control where the session setup time is improved when the circuit-switched bearer in the PSTN is described together with one or more codecs.

The rest of the document is structured as follows: Section 2 provides the document conventions, Section 3 introduces the requirements, Section 4 presents an overview of the proposed solutions, and Section 5 contains the protocol description. Section 6 provides an example of descriptions of circuit-switched audio or video streams in SDP. Section 7 and Section 8 contain the IANA and Security considerations, respectively.

2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [RFC2119] and indicate requirement levels for compliant implementations.

3. Requirements

This section presents the general requirements that are specific for the audio or video media streams over circuit-switched bearers.

REQ-1: A mechanism for endpoints to negotiate and agree on an audio or video media stream established over a circuit-switched bearer MUST be available.

REQ-2: The mechanism MUST allow the endpoints to combine circuit-switched audio or video media streams with other complementary media streams, for example, text messaging.

REQ-3: The mechanism MUST allow the endpoint to negotiate the direction of the circuit-switched connection, i.e., which endpoint is active when initiating the circuit-switched connection.

REQ-4: The mechanism MUST be independent of the type of the circuit-switched access (e.g., Integrated Services Digital Network (ISDN), Global System for Mobile Communication (GSM), etc.)

REQ-5: There MUST be a mechanism that helps an endpoint to correlate an incoming circuit-switched bearer with the one negotiated in SDP, as opposed to another incoming call that is not related to that.

REQ-6: It MUST be possible for endpoints to advertise different list of audio or video codecs in the circuit-switched audio or video stream from those used in a packet-switched audio or video stream.

REQ-7: It MUST be possible for endpoints to not advertise the list of available codecs for circuit-switched audio or video streams.

4. Overview of Operation

The mechanism defined in this memo extends SDP and allows describing an audio or video media stream established over a circuit-switched bearer. New tokens are registered in the "c=" and "m=" lines to be able to describe a media stream over a circuit-switched bearer. These SDP extensions are described in Section 5.2. Since circuit-switched bearers are connection-oriented media streams, the mechanism re-uses the connection-oriented extensions defined in RFC 4145 [RFC4145] to negotiate the active and passive sides of a connection setup. This is further described in Section 5.3.1.

4.1. Example Call Flow

Consider the example presented in Figure 1. In this example, Alice is located in an environment where she has access to both IP and circuit-switched bearers for communicating with other endpoints. Alice decides that the circuit-switched bearer offers a better perceived quality of service for voice, and issues an SDP Offer containing the description of an audio media stream over circuit-switched bearer.

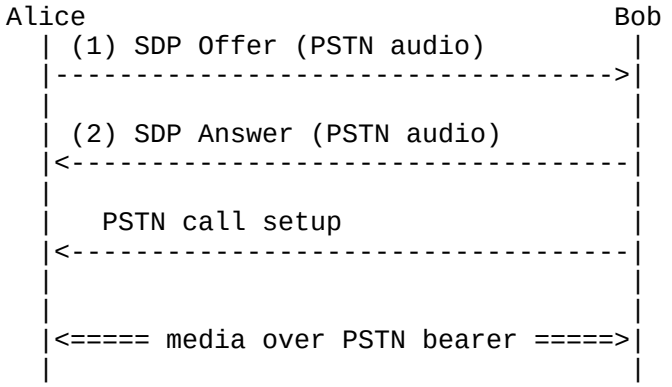


Figure 1: Example Flow

Bob receives the SDP offer and determines that he is located in an environment where the IP based bearer is not suitable for real-time audio media. However he also has PSTN circuit-switched bearer available for audio. Bob generates an SDP answer containing a description of the audio media stream over a circuit-switched bearer.

During the offer-answer exchange Alice and Bob also agree the direction in which the circuit-switched connection should be established. In this example, Bob becomes the active party, in other words, he establishes the circuit-switched call to the other endpoint. The Offer/Answer exchange contains identifiers or references that can be used on the circuit-switched network for addressing the other endpoint, as well as information that is used to determine that the incoming circuit-switched bearer establishment is related to the ongoing session between Alice and Bob.

Bob establishes a circuit-switched bearer towards Alice using whatever mechanisms are defined for the network type in question. When receiving the incoming circuit-switched connection attempt, Alice is able to determine that the attempt is related to the session she is just establishing with Bob.

Alice accepts the circuit-switched connection; the circuit-switched bearer setup is completed. Bob and Alice can now use the circuit-switched connection for two-way audio media.

If, for some reason, Bob would like to reject the offered stream, he would set the port number of the specific stream to zero, as specified in RFC3264 [RFC3264]. Also, if Bob does not understand some of the SDP attributes specified in this document, he would ignore them, as specified in RFC4566 [RFC4566].

5. Protocol Description

5.1. Level of Compliance

Implementations according to this specification MUST implement the SDP extensions described in Section 5.2, and MUST implement the considerations discussed in Section 5.3 and Section 5.4.

5.2. Extensions to SDP

This section provides the syntax and semantics of the extensions required for providing a description of audio or video media streams over circuit-switched bearers in SDP.

5.2.1. Connection Data

According to SDP [RFC4566], the connection data line in SDP has the following syntax:

```
c=<nettype> <addrtype> <connection-address>
```

where <nettype> indicates the network type, <addrtype> indicates the address type, and the <connection-address> is the connection address, which is dependent on the address type.

At the moment, the only network type defined is "IN", which indicates Internet network type. The address types "IP4" and "IP6" indicate the type of IP addresses.

This memo defines a new network type for describing a circuit-switched bearer network type in the PSTN. The mnemonic "PSTN" is used for this network type.

For the address type, we initially consider the possibility of describing E.164 telephone numbers. We define a new "E164" address type. When used, the "E164" address type indicates that the connection address contains a telephone number represented according to the ITU-T E.164 [ITU.E164.1991] recommendation.

There are cases, though, when the endpoint is merely aware of a circuit-switched bearer, without having further information about the address type or the E.164 number allocated to it. In these cases a dash "-" is used to indicate an unknown address type or connection address. This makes the connection data line be according to the SDP syntax.

Note that <addrtype> and/or <connection-address> should not be omitted without being set to a "-" since this would violate basic syntax of SDP [RFC4566].

The following are examples of the extension to the connection data line:

c=PSTN E164 +15551234

c=PSTN - -

5.2.2. Media Descriptions

According to SDP [RFC4566], the media descriptions line in SDP has the following syntax:

m=<media> <port> <proto> <fmt> ...

The <media> sub-field carries the media type. For establishing an audio bearer, the existing "audio" media type is used. For establishing a video bearer, the existing "video" media type is used.

The <port> sub-field is the transport port to which the media stream is sent. Circuit-switched access lacks the concept of a port number, and therefore the <port> sub-field is set to the discard port "9".

According to RFC 3264 [RFC3264], a port number of zero in the offer of a unicast stream indicates that the stream is offered but must not be used. If a port number of zero is present in the answer of a unicast stream, it indicates that the stream is rejected. These rules are still valid when the media line in SDP represents a circuit-switched bearer.

The <proto> sub-field is the transport protocol. The circuit-switched bearer uses whatever transport protocol it has available. This subfield SHOULD be set to the mnemonic "PSTN" to be syntactically correct with SDP [RFC4566] and to indicate the usage of circuit-switched protocols in the PSTN.

The <fmt> sub-field is the media format description. In the classical usage of SDP to describe RTP-based media streams, when the <proto> sub-field is set to "RTP/AVP" or "RTP/SAVP", the <fmt> sub-field contains the payload types as defined in the RTP audio profile [RFC3551].

When "RTP/AVP" is used in the <proto> field, the <fmt> sub-field contains the RTP payload type numbers. We use the <fmt> sub-field to indicate the list of available codecs over the circuit-switched

bearer, by re-using the conventions and payload type numbers defined for RTP/AVP. The RTP audio and video media types, which, when applied to PSTN circuit-switched bearers, represent merely an audio or video codec.

In some cases, the endpoint is not able to determine the list of available codecs for circuit-switched media streams. In this case, in order to be syntactically compliant with SDP [RFC4566], the endpoint MUST include a single dash "-" in the <fmt> sub-field.

As per RFC 4566 [RFC4566], the media format descriptions are listed in priority order.

Examples of media descriptions for circuit-switched audio streams are:

```
m=audio 9 PSTN 3 0 8

m=audio 9 PSTN -
```

Similarly, an example of a media description for circuit-switched video stream is:

```
m=video 9 PSTN 34

m=video 9 PSTN -
```

5.2.3. Correlating the PSTN Circuit-Switched Bearer with SDP

The endpoints should be able to correlate the circuit-switched bearer with the session negotiated with SDP in order to avoid ringing for an incoming circuit-switched bearer that is related to the session controlled with SDP (and SIP).

Several alternatives exist for performing this correlation. This memo provides three mutually non-exclusive correlation mechanisms. Other correlation mechanisms may exist, and their usage will be specified when need arises. All mechanisms share the same principle: some unique information is sent in the SDP and in the circuit-switched signaling protocol. If these pieces of information match, then the circuit-switched bearer is part of the session described in the SDP exchange. Otherwise, there is no guarantee that the circuit-switched bearer is related to such session.

The first mechanism is based on the exchange of PSTN caller-ID between the endpoints. The caller-ID is also available as the Calling Party ID in the circuit-switched signaling.

The second mechanism is based on the inclusion in SDP of a value that is also sent in the User-to-User Information Element that is part of the bearer setup signaling in the PSTN.

The third mechanism is based on sending in SDP a string that represents Dual Tone MultiFrequency (DTMF) digits that will be later sent right after the circuit-switched bearer is established. Implementations MAY use any of these mechanisms and MAY use two or more mechanisms simultaneously.

5.2.3.1. The "cs-correlation" attribute

In order to provide support for the correlation mechanisms, we define a new SDP attribute called "cs-correlation". This "cs-correlation" attribute can include any of the "callerid", "uuie", or "dtmf" sub-fields, which specify additional information required by the Caller-ID, User to User Information, or DTMF correlation mechanisms, respectively. The list of correlation mechanisms may be extended by other specifications.

The following sections provide more detailed information of these subfields. The "cs-correlation" attribute has the following format:

```
a=cs-correlation: callerid:<callerid-value> |
                  uuie:<uuie-value>         |
                  dtmf:<dtmf-value>         |
                  [extension-name:<extension-value>]
```

The values "callerid", "uuie" and "dtmf" refer to the correlation mechanisms defined in Section 5.2.3.2, Section 5.2.3.3, and Section 5.2.3.4, respectively. The formal Augmented Backus-Naur Format (ABNF) syntax of the "cs-correlation" attribute is presented in Section 5.5.

5.2.3.2. Caller-ID Correlation Mechanism

The Caller-ID correlation mechanisms consists of an exchange of the calling party number in E.164 format in SDP, followed by the availability of the Calling Party Number information element in the call setup signaling of the circuit switched connection. If both pieces of information match, the circuit-switched bearer is correlated to the session described in SDP.

Example of inclusion of E.164 number in the "cs-correlation" attribute is:

```
a=cs-correlation:callerid:+15551234
```

The presence of the "callerid" sub-field indicates that the endpoint supports use of the calling party number as a means of correlating a PSTN call with the session being negotiated. The "callerid" sub-field MAY be accompanied by the E.164 number of the party inserting the parameter. For details on negotiating the correlation mechanisms, see Section 5.3.

Note that there are no warranties that this correlation mechanism works or is even available, due a number of problems:

* The endpoint might not be aware of its own E.164 number, in which case it cannot populate the SDP appropriately.

* The Calling Party Number information element in the circuit-switched signaling might not be available, e.g., due to policy restrictions of the network operator or caller restriction due to privacy.

* The Calling Party Number information element in the circuit-switched signaling might be available, but the digit representation of the E.164 number might differ from the one expressed in the SDP. For example, one can be represented in international format and the other might only contain the significant national digits. To mitigate this problem implementations should consider only some of the rightmost digits from the E.164 number for correlation. For example, the numbers +358-1-555-12345 and 01-555-12345 could be considered as the same number. This is also the behavior of some cellular phones, which correlate the incoming calling party with a number stored in the phone book, for the purpose of displaying the caller’s name.

5.2.3.3. User-User Information Element Correlation Mechanism

A second correlation mechanism is based on including in SDP a string that represents the User-User Information Element that is part of the call setup signaling of the circuit-switched bearer. The User-User Information Element is specified in ITU-T Q.931 [ITU.Q931.1998] and 3GPP TS 24.008 [3GPP.24.008], among others. The User-User Information Element has a maximum size of 35 or 131 octets, depending on the actual message of the PSTN protocol where it is included.

The mechanism works as follows: An endpoint creates a User-User Information Element, according to the requirements of the call setup

signaling protocol. The same value is included in the SDP offer or SDP answer, in a "cs-correlation:uuie" attribute. When the SDP Offer/Answer exchange is completed, each endpoint has become aware of the value that will be used in the User-User Information Element of the call setup message of the PSTN protocol. The endpoint that initiates the call setup attempt includes this value in the User-User Information Element. The recipient of the call setup attempt can extract the User-User Information Element and correlate it with the value previously received in the SDP. If both values match, then the call setup attempt corresponds to that indicated in the SDP.

Note that, for correlation purposes, the value of the User-User Information Element is considered as a opaque string and only used for correlation purposes. Typically call signaling protocols impose requirements on the creation of User-User Information Element for end-user protocol exchange. The details regarding the generation of the User-User Information Element are outside the scope of this specification.

Please note that there are no warranties that this correlation mechanism works. On one side, policy restrictions might not make the User-User information available end to end in the PSTN. On the other hand, the generation of the User-User Information Element is controlled by the PSTN circuit-switched call protocol, which might not offer enough freedom for generating different values from one endpoint to another one, or from one call to another in the same endpoint. This might result in the same value of the User-User Information Element for all calls.

5.2.3.4. DTMF Correlation Mechanism

We introduce a third mechanism for correlating the circuit-switched bearer with the session described with SDP. This is based on agreeing on a sequence of digits that are negotiated in the SDP Offer/Answer exchange and sent as Dual Tone Multifrequency (DTMF) tones over the circuit-switched bearer once this bearer is established. If the DTMF digit sequence received through the circuit-switched bearer matches the digit string negotiated in the SDP, the circuit-switched bearer is correlated with the session described in the SDP. The mechanism is similar to many voice conferencing systems which require the user to enter a PIN code using DTMF tones in order to be accepted in a voice conference.

The mechanism works as follows: An endpoint selects a DTMF digit sequence. The same sequence is included in the SDP offer or SDP answer, in a "cs-correlation:dtmf" attribute. When the SDP Offer/ Answer exchange is completed, each endpoint has become aware of the DTMF sequence that will be sent right after the circuit-switched

bearer is set up. The endpoint that initiates the call setup attempt sends the DTMF digits according to the procedures defined for the circuit-switched bearer technology used. The recipient (passive side of the bearer setup) of the call setup attempt collects the digits and compares them with the value previously received in the SDP. If the digits match, then the call setup attempt corresponds to that indicated in the SDP.

Implementations are advised to select a number of DTMF digits that provide enough assurance that the call is related, but on the other hand do not prolong the bearer setup time unnecessarily.

As an example, an endpoint willing to send DTMF tone sequence "14D*3" would include a "cs-correlation" attribute line as follows:

```
a=cs-correlation:dtmf:14D*3
```

If the endpoints successfully agree on the usage of the DTMF digit correlation mechanism, but the passive side does not receive any DTMF digits after successful circuit-switched bearer setup, or receives a set of DTMF digits that do not match the value of the "dtmf" attribute (including receving too many digits), the passive side SHOULD treat the circuit-switched bearer as not correlated to the ongoing session.

DTMF digits can only be sent once the circuit-switched bearer is set up. In order to suppress alerting for an incoming circuit-switched call, implementations may choose various mechanisms. For example, alerting may be suppressed for a certain time period for incoming call attempts that originate from the number that was observed during the offer/answer negotiation.

5.3. Negotiating the correlation mechanisms

The three correlation mechanisms presented above (based on called party number, User-User Information Element and DTMF digit sending) are non-exclusive, and can be used independently of each other.

In order to agree which correlation mechanisms are supported and used by each endpoint, we define a negotiation mechanism similar to the one defined for codec negotiation. The sections below describe active/passive party determination and Offerer and Answerer behaviour for negotiating the correlation mechanisms.

5.3.1. Determining the Direction of the Circuit-Switched Connection Setup

In order to avoid a situation where both endpoints attempt to

initiate a connection simultaneously, the direction in which the circuit-switched bearer is set up should be negotiated during the Offer/Answer exchange.

The framework defined in RFC 4145 [RFC4145] allows the endpoints to agree which endpoint acts as the active endpoint when initiating a TCP connection. While RFC 4145 [RFC4145] was originally designed for establishing TCP connections, it can be easily extrapolated to the connection establishment of circuit-switched bearers. This specification uses the concepts specified in RFC 4145 [RFC4145] for agreeing on the direction of establishment of a circuit-switched bearer.

RFC 4145 [RFC4145] defines two new attributes in SDP: "setup" and "connection". The "setup" attribute indicates which of the endpoints should initiate the connection establishment of the PSTN circuit-switched bearer. Four values are defined in Section 4 of RFC 4145 [RFC4145]: "active", "passive", "actpass", "holdconn". Please refer to Section 4 of RFC 4145 [RFC4145] for a detailed description of this attribute.

The "connection" attribute indicates whether a new connection is needed or an existing connection is reused. The attribute can take the values "new" or "existing". Please refer to Section 5 of RFC 4145 [RFC4145] for a detailed description of this attribute.

Implementations according to this specification MUST support the "setup" and "connection" attributes specified in RFC 4145 [RFC4145], but applied to circuit-switched bearers in the PSTN.

We define the active party as the one that initiates the circuit-switched call after the Offer/Answer process. The passive party is the one receiving the circuit-switched call. Either party may indicate its desire to become the active or passive party during the Offer/Answer exchange using the procedures described below.

In order to establish a circuit-switched connection in the PSTN, the initiating endpoint needs to know the address (E.164 number) of the other endpoint. Therefore, if an endpoint wants to be able to receive incoming circuit-switched calls (i.e., become the passive party), it must know its E.164 number and must indicate it in SDP. As a consequence, an endpoint that is not aware of its own E.164 number cannot take the role of the passive side with respect the establishment of the circuit-switched connection.

5.3.2. Offerer behaviour

When generating the Offer, If the Offerer supports any of the correlation mechanisms defined in this memo, it SHOULD include an attribute line "a=cs-correlation" in the SDP offer. The "a=cs-correlation" line contains an enumeration of the correlation mechanisms supported by the Offerer, in the format of sub-fields.

The current list of sub-fields include "callerid", "uuie" and "dtmf" and they refer to the correlation mechanisms defined in Section 5.2.3.2, Section 5.2.3.3, and Section 5.2.3.4, respectively.

If the Offerer is only able to become the active party (for example because it doesn't know its E.164 address), the Offerer SHOULD add the "callerid", "uuie", and/or "dtmf" sub-fields but MUST NOT specify a value for those sub-fields.

If the Offerer is able to become the passive party in the circuit-switched call setup, it SHOULD add values to all correlation mechanisms it supports:

- the E.164 number as the value in the "callerid" sub-field,
- the contents of the User-User information element as the value of the "uuie" sub-field, and
- the DTMF tone string as the value of the "dtmf" sub-field

For example, if the Offerer is willing to use the User-User Information element and DTMF digit sending mechanisms, but can only become the active party, it includes the following line to the SDP:

```
a=cs-correlation:uuie dtmf
```

If, on the other hand, the Offerer is willing to use the User-User Information element and the DTMF correlation mechanisms, and is able to become the passive side, it includes to following line to the SDP:

```
a=cs-correlation:uuie:2890W284hAT452612908awudfjang908 dtmf:14D*3
```

When receiving the Answer, if the SDP does not contain "a=cs-correlation" attribute line, the Offerer should take that as an indication that the other party does not support or is not willing to use the procedures defined in the document for this session, and MUST revert to normal processing of SDP.

When receiving the Answer, the Offerer MUST first determine whether it becomes the active or passive party, as described in Section

Section 5.3.1.

If the Offerer becomes the active party, it

- SHOULD extract the E.164 address from the "c=" line and SHOULD establish a circuit-switched call to that address.
- if the SDP Answer contained a value for the "uuie" sub-field, SHOULD send the User-User Information element according to the rules defined for the circuit-switched technology used, and set the value of the Information Element to that received in the SDP Answer,
- if the SDP Answer contained a value for the "dtmf" sub-field, SHOULD send those DTMF digits according to the circuit-switched technology used.

If the Offerer becomes the passive party, it

- MUST be prepared to receive a circuit-switched call,
- MUST be prepared to receive and collect DTMF digits once the circuit-switched bearer is set up. The Offerer SHOULD compare the received DTMF digits to the value of the "dtmf" sub-field. If the received DTMF digits match the value of the "dtmf" sub-field in the "cs-correlation" attribute, the call SHOULD be treated as correlated to the ongoing session.
- MUST be prepared to receive a User-User Information element once the circuit-switched bearer is set up. The Offerer SHOULD compare the received UUI to the value of the "uuie" sub-field. If the value of the received UUI matches the value of the "uuie" sub-field, the call SHOULD be treated as correlated to the ongoing session.

5.3.3. Answerer behaviour

When receiving the Offer, the Answerer MUST first determine whether it becomes the active or passive party, as described in Section Section 5.3.1.

If the SDP in the Offer indicates that the Offerer is only able to become the active party ("a=setup" line is set to "active", the Answerer needs to determine whether it is able to become the passive party. If this is not possible e.g. due to the Answerer not knowing its E.164 address, the Answerer MUST NOT include "a=setup", "a=connection" or "a=cs-correlation" attributes in the Answer.

When generating the answer, the Answerer SHOULD select those correlation mechanisms from the Offer it supports, and include an "a=cs-correlation" attribute line in the answer containing those mechanisms it supports. The Answerer MUST NOT add any mechanisms which were not included in the offer.

If the Answerer becomes the active party, it MUST NOT add parameter values to the "callerid", "uuie" or "dtmf" sub-fields.

If the Answerer becomes the passive party, it MUST add values to the "callerid", "uuie" and/or "dtmf" sub-fields.

After generating and sending the Answer, if the Answerer became the active party, it

- SHOULD extract the E.164 address from the "c=" line of the Offer and SHOULD establish a circuit-switched call to that address.
- if the SDP Offer contained a value for the "uuie" sub-field, SHOULD send the User-User Information element according to the rules defined for the circuit-switched technology used, and set the value of the Information Element to that received in the SDP Offer,
- if the SDP Offer contained a value for the "dtmf" sub-field, SHOULD send those DTMF digits according to the circuit-switched technology used

If, on the other hand, the Answerer became the passive party, it

MUST be prepared to receive a circuit-switched call,

MUST be prepared to receive and collect DTMF digits once the circuit-switched bearer is set up. The Answerer SHOULD compare the received DTMF digits to the value of the "dtmf" sub-field. If the received DTMF digits match the value of the "dtmf" sub-field in the "cs-correlation" attribute, the call SHOULD be treated as correlated to the ongoing session.

MUST be prepared to receive a User-User Information element once the circuit-switched bearer is set up. The Answerer SHOULD compare the received UUI to the value of the "uuie" sub-field. If the value of the received UUI matches the value of the "uuie" sub-field, the call SHOULD be treated as correlated to the ongoing session.

5.3.4. Considerations on successful correlation

Note that, as stated above, it cannot be guaranteed that any given correlation mechanism will succeed even if the usage of those was agreed beforehand. This is due to the fact that the correlation mechanisms require support from the circuit-switched bearer technology used.

Therefore, even a single positive indication using any of these mechanisms SHOULD be interpreted by the passive endpoint so that the circuit-switched bearer establishment is related to the ongoing session, even if the other correlation mechanisms fail.

If, after negotiating one or more correlation mechanisms in the SDP offer/answer exchange, an endpoint receives a circuit-switched call with no correlation information present, the endpoint has two choices: it can either treat the call as unrelated, or treat the call as related to the ongoing session in the IP domain.

An endpoint may for example specify a time window after SDP offer/answer exchange during which received calls are treated as correlated even if the signaling in the circuit-switched domain does not carry any correlation information. In this case, there is a chance that the call is erroneously treated as related to the ongoing session.

An endpoint may also choose to always treat an incoming call as unrelated if the signaling in the circuit-switched domain does not carry any correlation information. In this case, there is a chance that the call is erroneously treated as unrelated.

Since, in these cases, no correlation information can be deduced from the signaling, it is up to the implementation to decide how to behave. One option is also to let the user decide whether to accept the call as related, or to treat the call as unrelated.

5.4. Considerations for Usage of Existing SDP

5.4.1. Originator of the Session

According to SDP [RFC4566], the origin line in SDP has the following syntax:

```
o=<username> <sess-id> <sess-version> <nettype> <addrtype>
<unicast-address>
```

Of interest here are the <nettype> and <addrtype> fields, which indicate the type of network and type of address, respectively. Typically, this field carries the IP address of the originator of the

session. Even if the SDP was used to negotiate an audio or video media stream transported over a circuit-switched bearer, the originator is using SDP over an IP bearer. Therefore, <nettype> and <addrtype> fields in the "o=" line should be populated with the IP address identifying the source of the signaling.

5.4.2. Contact information

SDP [RFC4566] defines the "p=" line which may include the phone number of the person reponsible for the conference. Even though this line can carry a phone number, it is not suited for the purpose of defining a connection address for the media. Therefore, we have selected to define the PSTN specific connection addresses in the "c=" line.

5.5. Formal Syntax

The following is the formal Augmented Backus-Naur Form (ABNF) [RFC5234] syntax that supports the extensions defined in this specification. The syntax is built above the SDP [RFC4566] grammar. Implementations according to this specification MUST be compliant with this syntax.

Figure 2 shows the formal syntax of the extensions defined in this memo.

```
; extension to the connection field originally specified
; in RFC 4566

connection-field   =  [%x63 "=" nettype SP addrtype SP
connection-address CRLF]
;nettype and addrtype are defined in RFC 4566

connection-address /=  e164-address / "-"
e164-address       =  ["+"] 1*15DIGIT
; DIGIT is specified in RFC 5234

;subrules for correlation attribute
attribute          /= cs-correlation-attr
; attribute defined in RFC 4566
cs-correlation-attr= "cs-correlation:" corr-mechanisms
corr-mechanisms    = corr-mech *(SP corr-mech)
corr-mech          = caller-id-mech / uuie-mech / dtmf-mech / ext-mech
caller-id-mech     = "callerid" [":" caller-id-value]
caller-id-value    = ["+"] 1*DIGIT
uuie-mech          = "uuie" [":" uuie-value]
uuie-value         = 1*32(ALPHA/DIGIT)
dtmf-mech          = "dtmf" [":" dtmf-value]
dtmf-value         = 1*32(DIGIT / %x41-44 / %x23 / %x2A )
;0-9, A-D, '#' and '*'
ext-mech           = ext-mech-name[":" ext-mech-value]
ext-mech-name      = token
ext-mech-value     = token
; token is specified in RFC4566
```

Figure 2: Syntax of the SDP extensions

6. Example

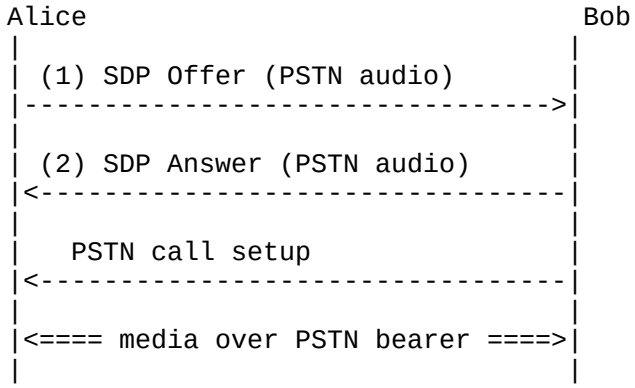


Figure 3: Basic flow

Figure 3 shows a basic example that describes a single audio media stream over a circuit-switched bearer. The SDP offer is show in Figure 4. The endpoint describes a PSTN circuit-switched bearer in the "m=" and "c=" line where it also indicates its E.164 number. Additionally, it expresses that it can initiate the circuit-switched connection or be the recipient of it. The SDP offer also includes a correlation identifier that this endpoint will be inserting the User-User Information Element of the PSTN call setup if eventually this endpoint initiates the PSTN call.

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 192.0.2.5
s=
t=0 0
m=audio 9 PSTN -
c=PSTN E164 +15551234
a=setup:actpass
a=connection:new
a=cs-correlation:uuie:2890W284hAT452612908awudfjang908
```

Figure 4: SDP offer (1)

7. IANA Considerations

This document instructs IANA to register a number of SDP tokens according to the following data.

7.1. Registration of new correlation SDP attribute

Contact: Miguel Garcia <miguel.a.garcia@ericsson.com>

Attribute name: cs-correlation

Long-form attribute name: PSTN Correlation Identifier

Type of attribute: media level only

This attribute is subject to the charset attribute

Description: This attribute provides the Correlation Identifier used in PSTN signaling

Specification: RFC XXXX

7.2. Registration of a new "nettype" value

This memo provides instructions to IANA to register a new "nettype" in the Session Description Protocol Parameters registry [1]. The registration data, according to RFC 4566 [RFC4566] follows.

Type	SDP Name	Reference
nettype	PSTN	[RFCxxxx]

7.3. Registration of new "addrtype" values

This memo provides instructions to IANA to register a new "addrtype" in the Session Description Protocol Parameters registry [1]. The registration data, according to RFC 4566 [RFC4566] follows.

Type	SDP Name	Reference
addrtype	E164	[RFCxxxx]
	-	[RFCxxxx]

7.4. Registration of a new "proto" value

This memo provides instructions to IANA to register a new "proto" in the Session Description Protocol Parameters registry [1]. The registration data, according to RFC 4566 [RFC4566] follows.

Type	SDP Name	Reference
proto	PSTN	[RFCxxxx]

8. Security Considerations

This document provides an extension on top of RFC 4566 [RFC4566], and RFC 3264 [RFC3264]. As such, the security considerations of those documents apply.

This memo provides mechanisms to agree on a correlation identifier or identifiers that are used to evaluate whether an incoming circuit-switched call is related to an ongoing session in the IP domain. If an attacker replicates the correlation identifer and establishes a call within the time window the receiving endpoint is expecting a call, the attacker may be able to hijack the circuit-switched call. These types of attacks are not specific to the mechanisms presented in this memo. For example, caller ID spoofing is a well known attack in the PSTN. Users are advised to use the same caution before revealing sensitive information as they would on any other phone

call. Furthermore, users are advised that mechanisms that may be in use in the IP domain for securing the media, like Secure RTP (SRTP) [RFC3711], are not available in the CS domain.

9. Acknowledgments

The authors want to thank Flemming Andreasen, Thomas Belling, John Elwell, Jari Mutikainen, Miikka Poikselka, Jonathan Rosenberg, Ingemar Johansson, Christer Holmberg, and Alf Heidermark for providing their insight and comments on this document.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3108] Kumar, R. and M. Mostafa, "Conventions for the use of the Session Description Protocol (SDP) for ATM Bearer Connections", RFC 3108, May 2001.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

[RFC4145] Yon, D. and G. Camarillo, "TCP-Based Media Transport in the Session Description Protocol (SDP)", RFC 4145, September 2005.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

10.2. Informative References

[3GPP.24.008] 3GPP, "Mobile radio interface Layer 3 specification; Core network protocols; Stage 3", 3GPP TS 24.008 3.20.0, December 2005.

[ITU.E164.1991] International Telecommunications Union, "The International Public Telecommunication Numbering Plan", ITU-

T Recommendation E.164, 1991.

[ITU.Q931.1998]
"Digital Subscriber Signalling System No. 1 (DSS 1) - ISDN User - Network Interface Layer 3 Specification for Basic Call Control", ISO Standard 9594-1, May 1998.

[RFC3261] 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.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, July 2003.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[RFC4975] Campbell, B., Mahy, R., and C. Jennings, "The Message Session Relay Protocol (MSRP)", RFC 4975, September 2007.

URIs

[1] <http://www.iana.org/assignments/sdp-parameters>

Authors' Addresses

Miguel A. Garcia-Martin
Ericsson
Calle Via de los Poblados 13
Madrid, ES 28033
Spain

Email: miguel.a.garcia@ericsson.com

Simo Veikkolainen
Nokia
P.O. Box 407
NOKIA GROUP, FI 00045
Finland

Phone: +358 50 486 4463
Email: simo.veikkolainen@nokia.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 15, 2011

C. Jennings
A. Begen
Cisco

March 14, 2011

Grouping of Adjacent Media in the Session Description Protocol
draft-jennings-mmusic-adjacent-grouping-03

Abstract

Applications such as multi-screen video conferencing systems or advertisement boards often have multiple audio and video streams that are organized to be rendered side by side or in a grid. This specification uses the RFC 5888 Grouping Framework to define new semantics for grouping the media streams to be rendered side by side or in a grid and indicating their relative ordering.

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 September 15, 2011.

Copyright Notice

Table of Contents

1. Introduction

There are many situations where applications create media streams that are meant do be rendered adjacent to each other. A common example is a multi-screen video conferencing system. Other examples are several video monitors placed side by side to display signs, and audio streams from a linear array of microphones, or a grid of display for monitoring security cameras. The Session Description Protocol (SDP) [RFC4566] allows negotiation of multiple media streams but does not have a way to describe the ordering information to indicate which media stream is adjacent to which one.

This specification introduces new grouping semantics, using the SDP Grouping Framework defined in [RFC5888], that indicate media streams are adjacent, and the adjacency order is defined by the order of the entries in the group.

2. Terminology

This specification uses all the terms defined in [RFC5888] and will not make sense unless you have read [RFC5888]. 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 [RFC2119].

3. Adjacent Media Grouping

3.1. "ADJ" Grouping Semantics

This specification defines new grouping semantics of "ADJ" that indicate the media streams in this group are meant to be played or displayed adjacently. Furthermore, the order of media streams in the group indicates the adjacency order. This only indicates the order the device sending the SDP believes is the preferred way to display the media described in this SDP. This is a declarative SDP parameter and is not negotiated.

N media streams could be in a linear horizontal layout, in which case we use a grid size of 1 x N. Alternatively, N media streams could be in a linear vertical layout, in which case we use a grid size of N x 1. In these configurations, the first stream in the group MUST be the one corresponding to the left most and top most output unit, respectively. In a more general grid size of N x M, we can group K (where K <= N x M) media streams starting from the one corresponding to the top-left output unit, and then doing a continuous horizontal scanning of the grid row by row (i.e., scanning first the top row

from left to right, and then the second row from left to right, and so on). When we say left most, we mean from the point of view of the person looking at the display.

To indicate the dimensions of the layout grid in an SDP description, we define a new session-level attribute. The ABNF syntax [RFC5234] for the new attribute is as follows:

```
media-grid-dims-line = "a=media-grid-dims:" rows "x" columns CRLF

rows    = %x31-39 *DIGIT
columns = %x31-39 *DIGIT
```

The parameters 'rows' and 'columns' indicate the number of rows and columns for this media grid. They both MUST be an integer larger than zero.

If the 'media-grid-dims' attribute does not exist in the SDP description, then a 1 x N horizontal linear layout MUST be assumed.

Per [RFC5888], there MAY be more than one adjacent media group in a single SDP description.

3.2. Grouping for SSRC-Multiplexed RTP Streams

[RFC5576] defines an SDP media-level attribute, called 'ssrc-group', for grouping the RTP streams that are SSRC multiplexed and carried in the same RTP session. The grouping is based on the SSRC identifiers. Since SSRC-multiplexed RTP streams are defined in the same "m" line, the 'group' attribute cannot be used.

This section specifies how adjacency is described with SSRC-multiplexed streams using the 'ssrc-group' attribute [RFC5576].

The semantics of "ADJ" for the 'ssrc-group' attribute are the same as the one defined for the 'group' attribute except that the SSRC identifiers are used to designate the adjacency grouping associations: a=ssrc-group:ADJ *(SP ssrc-id) [RFC5576].

The SSRC identifiers for the RTP streams that are carried in the same RTP session MUST be unique per [RFC3550]. However, the SSRC identifiers are not guaranteed to be unique among different RTP sessions. Thus, the 'ssrc-group' attribute MUST only be used at the media level [RFC5576].

3.3. SDP Offer/Answer Model Considerations

When offering adjacent media grouping using SDP in an Offer/Answer model [RFC3264], the following considerations apply.

A node that is receiving an offer from a sender may or may not understand line grouping. It is also possible that the node understands line grouping but it does not understand the "ADJ" semantics. From the viewpoint of the sender of the offer, these cases are indistinguishable.

When a node is offered a session with the "ADJ" grouping semantics but it does not support line grouping or the adjacent media grouping semantics, as per [RFC5888], the node responds to the offer either (1) with an answer that ignores the grouping attribute or (2) with a refusal to the request (e.g., 488 Not Acceptable Here or 606 Not Acceptable in SIP).

In the first case, the original sender of the offer must send a new offer without any grouping. In the second case, if the sender of the offer still wishes to establish the session, it should retry the request with an offer without the adjacent media grouping. This behavior is specified in [RFC5888].

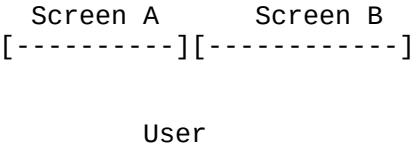
The offer MUST contain the sender’s desired layout. The answer MAY contain the desired layout of the streams that the system sending the answer will be sending to the system that sent the offer.

4. SDP Examples

This section provides SDP examples showing how to use the adjacent media grouping.

4.1. Horizontal Layout

A video system with two screens and one audio channels sends a SIP offer. The following figure shows a top-down view of the room with the three screen system that is sending the SIP offer. Screen A is the left most screen for the user in this room but should be displayed as the rightmost screen for the user at the far end that will be viewing the video.



Assume the SDP mid values for the screens are sa and sb, for Screens A and B respectively. The offer contains the following in the SDP:

```
a=group:ADJ sb sa
```

The complete SDP in the offer could look like:

```
v=0
o=alice 2890844526 2890844526 IN IP4 host.atlanta.example.com
s=
c=IN IP4 host.atlanta.example.com
a=group:ADJ sb sa
t=0 0
m=audio 49101 RTP/AVP 101
a=rtpmap:101 PCMU/8000
m=video 49111 RTP/AVP 111
a=rtpmap:111 H261/90000
a=mid:sa
m=video 49112 RTP/AVP 112
a=rtpmap:112 H261/90000
a=mid:sb
```

There might be other media streams, such as presentation video, that are not part of any "ADJ" group.

As a note to implementors, consider the case where each screen had two media flows that were in the same FID group. In this case all the media streams are still listed in the ADJ group and the order of two streams in the same FID group can be arbitrarily picked as they will be displayed on the same device.

4.2. Grid Layout

The following SDP is for a system providing 6 video streams to a wall of screens. The wall has 3 columns and 2 rows of screens.

```
v=0
o=bob 2890844526 2890844526 IN IP4 host.atlanta.example.com
s=
c=IN IP4 host.atlanta.example.com
a=group:ADJ 1 2 3 4 5 6
a=media-grid-dims:2x3
t=0 0
m=audio 49101 RTP/AVP 101
a=rtpmap:101 H261/90000
a=mid:1
m=audio 49102 RTP/AVP 102
a=rtpmap:102 H261/90000
a=mid:2
m=audio 49103 RTP/AVP 103
a=rtpmap:103 H261/90000
a=mid:3
m=audio 49104 RTP/AVP 104
a=rtpmap:104 H261/90000
a=mid:4
m=audio 49105 RTP/AVP 105
a=rtpmap:105 H261/90000
a=mid:5
m=audio 49106 RTP/AVP 106
a=rtpmap:106 H261/90000
a=mid:6
```

5. Security Considerations

Like all SDP, integrity of this information is important. When carrying SDP in SIP, mechanisms such as Transport Layer Security (TLS) can provide hop by hop confidentiality and integrity. The receiver SHOULD do an integrity check on SDP and follow the security considerations of SDP [RFC4566] to trust only SDP from trusted sources. End-to-end integrity can be provided by [RFC4474].

6. IANA Considerations

Note to RFC Editor: Please replace [RFC-AAAA] with the RFC number for this specification.

6.1. Registration of SDP Attributes

This document registers a new attribute name in SDP.

SDP Attribute ("att-field"):

Attribute name:	media-grid-dims
Long form:	2-D media grid dimensions
Type of name:	att-field
Type of attribute:	Session level
Subject to charset:	No
Purpose:	Specifies the dimensions for a media grid
Reference:	[RFC-AAAA]
Values:	See [RFC-AAAA]

6.2. Registration of Grouping Semantics

This document, following the Standards Action policy from [RFC5226], registers the following semantics with IANA in the "Semantics for the "group" SDP Attribute" registry under SDP Parameters:

Semantics	Token	Reference
Adjacent Media	ADJ	[RFC-AAAA]

This document also registers the following semantics with IANA in "Semantics for the 'ssrc-group' SDP Attribute" registry under SDP Parameters:

Token	Semantics	Reference
RED	Adjacent Media	[RFC-AAAA]

7. Acknowledgments

The authors would like to thank Flemming Andreasen, Allyn Romanow, Roni Even, Hakon Dahle, Ingemar Johansson, Peter Musgrave, and Geir Arne Sandbakken for their review comments.

8. References

8.1. Normative References

[RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description Protocol (SDP) Grouping Framework", RFC 5888, June 2010.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC5576] Lennox, J., Ott, J., and T. Schierl, "Source-Specific Media Attributes in the Session Description Protocol (SDP)", RFC 5576, June 2009.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

8.2. Informative References

[RFC4474] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, August 2006.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

Authors' Addresses

Cullen Jennings
Cisco
170 West Tasman Drive
San Jose, CA 95134
USA

Phone: +1 408 421-9990
Email: fluffy@cisco.com

Ali Begen
Cisco
181 Bay Street
Toronto, ON M5J 2T3
Canada

Email: abegen@cisco.com

MMUSIC
Internet-Draft
Intended status: Standards Track
Expires: September 15, 2011

S. Loreto
G. Camarillo
Ericsson

March 14, 2011

Stream Control Transmission Protocol (SCTP)-Based Media Transport in the Session Description Protocol (SDP)
draft-loreto-mmusic-sctp-sdp-07

Abstract

SCTP (Stream Control Transmission Protocol) is a transport protocol used to establish associations between two endpoints. This document describes how to express media transport over SCTP in SDP (Session Description Protocol). This document defines the 'SCTP' and 'SCTP/DTLS' protocol identifiers for SDP.

Status of this Memo

This Internet-Draft is submitted to IETF 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 September 15, 2011.

Copyright Notice

described in the Simplified BSD License.

Table of Contents

1. Introduction

SDP (Session Description Protocol) [RFC4566] provides a general-purpose format for describing multimedia sessions in announcements or invitations. RFC4145 [RFC4145] specifies a general mechanism for describing and establishing TCP (Transmission Control Protocol) streams. RFC 4572 [RFC4572] extends RFC4145 [RFC4145] for describing TCP-based media streams that are protected using TLS (Transport Layer Security) [RFC5246].

This document defines a new protocol identifier, 'SCTP', to describe SCTP-based [RFC4960] media streams. Additionally, this document specifies the use of the 'setup' and 'connection' SDP attributes to establish SCTP associations. These attributes were defined in RFC4145 [RFC4145] for TCP. This document discusses their use with SCTP.

Additionally this document defines a new protocol identifier, 'SCTP/DTLS', to establish secure SCTP-based media streams over DTLS (Datagram Transport Layer Security) [RFC4347], as specified in [RFC6083], using SDP. The authentication certificates are interpreted and validated as defined in RFC4572 [RFC4572]. Self-signed certificates can be used securely, provided that the integrity of the SDP description is assured as defined in RFC4572 [RFC4572].

TLS is designed to run on top of a byte-stream oriented transport protocol providing a realible, in-sequence delivery like TCP. Since no-one so far has implemented SCTP over TLS, due to some serious limitations described in [RFC6083], this document does not make use of TLS over SCTP as described in RFC3436 [RFC3436].

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 [RFC2119] and indicate requirement levels for compliant implementations.

3. Protocol Identifier

The following is the format for an 'm' line, as specified in RFC4566 [RFC4566]:

```
m=<media> <port> <proto> <fmt> ...
```

This document defines two new values for the 'proto' field: 'SCTP' and 'SCTP/DTLS'.

The 'SCTP' protocol identifier is similar to both the 'UDP' and 'TCP' protocol identifiers in that it only describes the transport protocol and not the upper-layer protocol. Media described using an 'm' line containing the 'SCTP' protocol identifier are carried using SCTP [RFC4960].

The 'SCTP/DTLS' protocol identifier indicates that the media described will use the Datagram Transport Layer Security (DTLS) [RFC4347] over SCTP as specified in [RFC6083].

An 'm' line that specifies 'SCTP' or 'SCTP/DTLS' MUST further qualify the application-layer protocol using an fmt identifier.

An 'm' line that specifies 'SCTP/DTLS' MUST further provide a certificate fingerprint. An SDP attribute (an 'a' line) is used to transport and exchange end point certificate. The authentication certificates are interpreted and validated as defined in [RFC4572].

4. The Setup and Connection Attributes and Association Management

The use of the 'setup' and 'connection' attributes in the context of an SCTP association is identical to the use of these attributes in the context of a TCP connection. That is, SCTP endpoints MUST follow the rules in Sections 4 and 5 of RFC 4145 [RFC4145] when it comes to the use of the 'setup' and 'connection' attributes in offer/answer [RFC3264] exchanges.

The management of an SCTP association is identical to the management of a TCP connection. That is, SCTP endpoints MUST follow the rules in Section 6 of RFC 4145 [RFC4145] to manage SCTP associations. Whether to use the SCTP ordered or unordered delivery service is up to the applications using the SCTP association.

5. Multihoming

An SCTP endpoint, unlike a TCP endpoint, can be multihomed. An SCTP endpoint is considered to be multihomed if it has more than one IP address. A multihomed SCTP endpoint informs a remote SCTP endpoint about all its IP addresses using the address parameters of the INIT or the INIT-ACK chunk (depending on whether or not the multihomed endpoint is the one initiating the establishment of the association). Therefore, once the address provided in the 'c' line has been used to establish the SCTP association (i.e., to send the INIT chunk),

address management is performed using SCTP. This means that two SCTP endpoints can use addresses that were not listed in the 'c' line but that were negotiated using SCTP mechanisms.

Some intermediaries performing firewall control use the addresses in offer/answer exchanges to perform media authorization. That is, they will not let media through unless it is sent to the address in the 'c' line.

The SCTP endpoints MAY choose to use the main address all the time (e.g., they do not send retransmissions to a backup address) and to send a re-INVITE every time they change that address.

However not using non-primary paths for retransmission means not to utilize the multi-homing feature of SCTP for resilience. Therefore, the SCTP endpoints MAY use the 'candidate' SDP attribute to disclosure, to intermediaries performing firewall control, all its alternative IP addresses; this will avoid the need for the SCTP endpoints to send a re-INVITE every time they change the primary path.

6. Examples

The following examples show the use of the 'setup' and 'connection' SDP attributes. As discussed in Section 4, the use of these attributes with an SCTP association is identical to their use with a TCP connection. 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).

6.1. Actpass/Passive

An offerer at 192.0.2.2 signals its availability for an SCTP association at SCTP port 54111. Additionally, this offerer is also willing to initiate the SCTP association:

```
m=image 54111 SCTP *
c=IN IP4 192.0.2.2
a=setup:actpass
a=connection:new
```

Figure 1

The endpoint at 192.0.2.1 responds with the following description:

```
m=image 54321 SCTP *
c=IN IP4 192.0.2.1
a=setup:passive
a=connection:new
```

Figure 2

This will cause the offerer (at 192.0.2.2) to initiate an SCTP association to port 54321 at 192.0.2.1.

6.2. Existing Connection Reuse

Subsequent to the exchange in Section 6.1, another offer/answer exchange is initiated in the opposite direction. The endpoint at 192.0.2.1, which now acts as the offerer, wishes to continue using the existing association:

```
m=application 54321 SCTP *
c=IN IP4 192.0.2.1
a=setup:passive
a=connection:new
```

Figure 3

The endpoint at 192.0.2.2 also wishes to use the existing SCTP association and responds with the following description:

```
m=application 9 SCTP *
c=IN IP4 192.0.2.2
a=setup:active
a=connection:new
```

Figure 4

The existing SCTP association between 192.0.2.2 and 192.0.2.1 will be reused.

6.3. SDP description for DTLS Connection

An offerer at 192.0.2.2 signals the availability of a T.38 fax session over SCTP/DTLS.

```
m=image 54111 SCTP/DTLS t38
c=IN IP4 192.0.2.2
a=setup:actpass
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
```

Figure 5

7. Network Address Translators (NAT) Considerations

SCTP specific features (not present in UDP/TCP), as the checksum (CRC32c) value calculated on the whole packet (not just the header) or multihoming, present new challenges for a NAT. [I-D.ietf-behave-sctpnat] describes an SCTP specific variant of NAT which provides similar features of Network Address and Port Translation (NAPT).

As an alternative to design SCTP specific NAT, SCTP-over-UDP [I-D.tuexen-sctp-udp-encaps] makes possible, encapsulating SCTP packets into UDP packets, to use SCTP in networks with legacy NAT and firewalls not supporting SCTP.

OPEN ISSUE: do we want to include in SDP the ability to signal SCTP-over-UDP ?

TBD: How to use Interactive Connectivity Establishment (ICE) to establish SCTP streams.

OPENT ISSUE: Do we want ICE to only use SCTP over IP candidates, or we ant ICE to use SCTP over UDP candidates as well?

8. Security Considerations

See RFC 4566 [RFC4566] for security considerations on the use of SDP in general. See RFC 3264 [RFC3264], RFC 4145 [RFC4145] and RFC 4572 [RFC4572] for security considerations on establishing media streams using offer/answer exchanges. See RFC 4960 [RFC4960] for security considerations on SCTP in general and [RFC6083] for security consideration using DTLS on top of SCTP. This specification does not introduce any new security consideration in addition to the ones discussed in those specifications.

9. IANA Considerations

This document defines two new proto values: 'SCTP' and 'SCTP/DTLS'. Their formats are defined in Section 3. These proto values should be registered by the IANA under "Session Description Protocol (SDP) Parameters" under "proto".

The SDP specification, [RFC4566], states that specifications defining new proto values, like the SCTP and SCTP/DTLS proto values defined in this RFC, must define the rules by which their media format (fmt) namespace is managed. For the SCTP protocol, new formats SHOULD have an associated MIME registration. Use of an existing MIME subtype for the format is encouraged. If no MIME subtype exists, it is RECOMMENDED that a suitable one is registered through the IETF process [RFC4288] [RFC4289] by production of, or reference to, a standards-track RFC that defines the transport protocol for the format.

10. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

[RFC3436] Jungmaier, A., Rescorla, E., and M. Tuexen, "Transport Layer Security over Stream Control Transmission Protocol", RFC 3436, December 2002.

[RFC4145] Yon, D. and G. Camarillo, "TCP-Based Media Transport in the Session Description Protocol (SDP)", RFC 4145, September 2005.

[RFC4288] Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", BCP 13, RFC 4288, December 2005.

[RFC4289] Freed, N. and J. Klensin, "Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures", BCP 13, RFC 4289, December 2005.

[RFC4347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", RFC 4347, April 2006.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC4572] Lennox, J., "Connection-Oriented Media Transport over the Transport Layer Security (TLS) Protocol in the Session Description Protocol (SDP)", RFC 4572, July 2006.

[RFC4960] Stewart, R., "Stream Control Transmission Protocol", RFC 4960, September 2007.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC6083] Tuexen, M., Seggelmann, R., and E. Rescorla, "Datagram Transport Layer Security (DTLS) for Stream Control Transmission Protocol (SCTP)", RFC 6083, January 2011.

[I-D.ietf-behave-sctpnat] Stewart, R., Tuexen, M., and I. Ruengeler, "Stream Control Transmission Protocol (SCTP) Network Address Translation", draft-ietf-behave-sctpnat-04 (work in progress), December 2010.

[I-D.tuexen-sctp-udp-encaps] Tuexen, M. and R. Stewart, "UDP Encapsulation of SCTP Packets", draft-tuexen-sctp-udp-encaps-06 (work in progress), January 2011.

Authors' Addresses

Salvatore Loreto
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: Salvatore.Loreto@ericsson.com

Gonzalo Camarillo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: Gonzalo.Camarillo@ericsson.com

MMUSIC WG
Internet-Draft
Intended status: Standards Track
Expires: September 9, 2011

M. Garcia-Martin
Ericsson
S. Veikkolainen
Nokia
R. Gilman

March 8, 2011

Title and Bandwidth Capabilities Negotiation in the Session Description Protocol (SDP)
draft-mmusic-sdp-icap-bcap-01

Abstract

SDP has been extended with a capability negotiation mechanism framework that allows the endpoints to negotiate transport protocols and attributes. This framework has been extended with a media capabilities negotiation mechanism that allows endpoints to negotiate additional media-related capabilities. This negotiation is embedded into the widely-used SDP offer/answer procedures.

This memo extends the SDP capability negotiation framework to allow endpoints to negotiate two additional SDP capabilities. In particular, this memo provides a mechanism to negotiate titles ("i=" line for each session or media) and bandwidth ("b=" line).

Table of Contents

1. Introduction

The Session Description Protocol (SDP) [RFC4566] is intended for describing multimedia sessions for the purposes of session announcement, session invitation, and other forms of multimedia session initiation. SDP has been extended with a capability negotiation mechanism framework [RFC5939] which allows the endpoints to negotiate capabilities, such as support for Real-time Transport Protocol (RTP) [RFC3550] and Secure Real-time Transport Protocol (SRTP) [RFC3711]. The SDP media capabilities [I-D.ietf-mmusic-sdp-media-capabilities] provides negotiation capabilities to media lines as well.

The capability negotiation is embedded into the widely used SDP offer/answer procedure [RFC3264]. This memo provides the means to negotiate further capabilities than those specified in the SDP capability negotiation mechanism framework [RFC5939] and the SDP media capabilities negotiation [I-D.ietf-mmusic-sdp-media-capabilities]. In particular, this memo provides a mechanism to negotiate session or media titles ("i=") and bandwidth ("b=").

Since the two added capabilities are highly unconnected, it is not expected that implementations will support both at the same time. Instead, it is expected that applications will choose their needed capability for their specific purpose. Due to this, we are writing the normative part pertaining to both capabilities in a self-contained section: Section 3.1.1 describes the bandwidth capability extension, and Section 3.1.2 describes the title capability extension. Separate option tags are defined for both capabilities.

2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [RFC2119] and indicate requirement levels for compliant implementations.

3. Protocol Description

3.1. Extensions to SDP

The SDP Capability Negotiation Framework [RFC5939] and the SDP media capabilities negotiation [I-D.ietf-mmusic-sdp-media-capabilities] specify attributes for negotiating SDP capabilities. These documents

specify new attributes (e.g., 'acap', 'tcap', 'mcap') for achieving their purpose. In this document we define two new additional capability attributes for SDP lines of the the general form:

```
type=value
```

for types "i" and "b". The corresponding capability attributes are defined as "icap" for title capability, and "bcap" for bandwidth capability, respectively.

From the sub-rules of "a=" line in SDP [RFC4566], SDP attributes are of the form:

```
attribute          = (att-field ":" att-value) / att-field
att-field          = token
att-value          = byte-string
```

Capability attributes use only the 'att-field:att-value' form.

The new attributes may be referenced in potential configurations ("a=pcfg") or in latent configurations ("a=lcfg"), as productions conforming to the extension-config-list as defined in [RFC5939].

```
extension-config-list = ["+"] ext-cap-name "=" ext-cap-list
ext-cap-name          = 1*(ALPHA / DIGIT)
                         ; ALPHA and DIGIT defined in RFC5234
ext-cap-list          = 1*VCHAR  ; VCHAR defined in RFC5234
```

The optional "+" is used to indicate that the extension is mandatory and MUST be supported in order to use that potential configuration.

The attributes may be referenced in actual configurations ("a=acfg") as productions conforming to the sel-extension-config defined in [RFC5939].

```
sel-extension-config = ext-cap-name "=" 1*VCHAR
```

The specific parameters are defined in the individual description of each capability, below.

The "icap" and "bcap" capability attributes can be provided either at the session or media level. According to the SDP Capability Negotiation [RFC5939], each extension capability must specify the implication of making it part of a configuration at the media level.

According to SDP [RFC4566], "b=" and "i=" lines may appear either at session or media level. In line with this, the "bcap" and "icap" capability attributes, when declared at session level, are to be

interpreted as-if that attribute was provided with that value at the session level. The "bcap" and "icap" capability attributes declared at media level, are to be interpreted as-if that capability attribute was declared at the session level.

For example, extending the example in [I-D.ietf-mmusic-sdp-media-capabilities] with "icap" and "bcap" capability attributes, we get the following SDP:

```
v=0
a=bcap:1 CT:200
a=icap:1 Video conference
m=audio 54320 RTP/AVP 0
a=mcap:1 L16/8000/1
a=mcap:2 L16/16000/2
a=pcfg:1 m=1|2, pt=1:99,2:98
m=video 66544 RTP/AVP 100
a=mcap:3,4 H263-1998/90000
a=rtpmap:100 H264/90000
a=pcfg:10 m=3 pt=3:101 b=1 i=1
```

Example SDP offer with bcap and icap defined at session level

The above SDP defines one PCMU audio stream and one H.264 video stream. It also defines two Media Format capabilities (numbered 1 and 2), using L16 audio at 8 kbps and 16 kbps, respectively, as well as Media Format capabilities for H.263 video (numbered 3 and 4). The Media Format capabilities all appear at the media level. The example also contains a single bandwidth capability and a single title capability at session level. According to the definition above, when the capabilities defined in "bcap", and "icap" attributes are referenced from the potential configuration, in the resulting SDP they are to be interpreted as session level attributes (but the Media Format capabilities are to be interpreted as media level attributes).

3.1.1. Bandwidth Capability

According to RFC 4566 [RFC4566] the bandwidth field denotes the proposed bandwidth to be used by the session or media. In this memo, we specify the bandwidth capability attribute which can also appear either at session or media level. The bandwidth field is specified in RFC 4566 [RFC4566] with the following syntax:

```
b=<bwtype>:<bandwidth>
```

where <bwtype> is an alphanumeric modifier giving the meaning of the <bandwidth> figure.

In this document, we define a new capability attribute: the bandwidth capability attribute "bcap". This attribute lists bandwidth as capabilities according to the following definition:

"a=bcap:" bw-cap-num 1*WSP bwtype ":" bandwidth CRLF

where <bw-cap-num> is a unique integer between 1 and 2^31-1 (both included) user to number the bandwidth capability, and the other elements are as defined for the "b=" field in SDP [RFC4566].

This format satisfies the general attribute production rules in SDP [RFC4566] according to the following Augmented Backus-Naur Form (ABNF) [RFC5234] syntax:

```
att-field       = "bcap"
att-value       = bw-cap-num 1*WSP bwtype ":" bandwidth
bw-cap-num      = 1*10(DIGIT)   ; DIGIT defined in RFC5234
```

Negotiation of bandwidth per media stream can be useful when negotiating media encoding capabilities with different bandwidths.

3.1.1.1. Configuration Parameters

The SDP capability negotiation framework [RFC5939] provides for the existence of the "pcfg" and "acfg" attributes. The concept is extended by the SDP media capabilities negotiation [I-D.ietf-mmusic-sdp-media-capabilities] with an "lcfg" attribute that conveys latent configurations.

Extensions to the "pcfg" and "lcfg" attributes are defined through <extension-config-list>, and extensions to the "acfg" attribute are defined through the <sel-extension-config> as defined in the SDP Capability Negotiation [RFC5939].

In this document we extend the <extension-config-list> field to be able to convey lists of bandwidth capabilities in latent or potential configurations, according to the following Augmented Backus-Naur Form (ABNF) [RFC5234] syntax:

```
extension-config-list  = bandwidth-config-list
bandwidth-config-list  = ["+"] "b=" bw-cap-list *(BAR bw-cap-list)
                         ; BAR defined in RFC5939
bw-cap-list            = bw-cap-num *("," bw-cap-num)
bw-cap-num             = 1*10(DIGIT)   ; DIGIT defined in RFC5234
```

Figure 1: Syntax of the bandwidth parameter in lcfg and pcfg attributes

Each bandwidth capability configuration is a comma-separated list of bandwidth capability attribute numbers where 'bw-cap-num' refers to the bw-cap-num bandwidth capability numbers defined explicitly earlier in this document, and hence must be between 1 and 2^31-1 (both included). Alternative bandwidth configurations are separated by a vertical bar ("|").

The above syntax is very flexible, allowing referencing to multiple "b=" lines per configuration, even for the same bwtype. While the need for such definitions is not seen, we have not restricted this, as it is not restricted in SDP [RFC4566] either.

The bandwidth parameter to the actual configuration attribute ("a=acfg") is formulated as a sel-extension-config with

```
ext-cap-name = "b"
```

hence

```
sel-extension-config = sel-bandwidth-config
sel-bandwidth-config = "b=" bw-cap-list  ; bw-cap-list as above.
```

Figure 2: Syntax of the bandwidth parameter in acfg attributes

3.1.1.2. Option tag

The SDP Capability Negotiation Framework [RFC5939] allows for capability negotiation extensions to be defined. Associated with each such extension is an option tag that identifies the extension in question. Hereby, we define a new option tag "bcap-v0" that identifies support for the bandwidth capability. The endpoints using the "bcap" capability attribute SHOULD add the option tag to other existing option tags present in the "csup" and "creq" attributes in SDP, according to the procedures defined in the SDP Capability Negotiation Framework [RFC5939].

3.1.2. Title Capability

SDP [RFC4566] provides for the existence of an information field expressed in the format of the "i=" line, which can appear either at the session level or at the media level. An "i=" line that is present at the session level is known as the "session name", and its purpose is to convey a human-readable textual information about the session.

The "i=" line in SDP can also appear at the media level, in which case it is used to provide human-readable information about the media stream to which it is related, e.g., it may indicate the purpose of

the media stream. The "i=" line is not to be confused with the label attribute ("a=label:", [RFC4574]) which provides a machine-readable tag. It is foreseen that applications declaring capabilities related to different configurations of a media stream may need to provide different identifying information for each of those configurations. That is, a party might offer alternative media configurations for a stream, each of which represents a different presentation of the same or similar information. For example, an audio stream might offer English or Spanish configurations, or a video stream might offer a choice of video source such as speaker camera, group camera, or document viewer. The title capability is needed to inform the answering user in order to select the proper choice, and the label is used to inform the offering machine which choice the answerer has selected. Hence, there is value in defining a mechanism to provide titles of media streams as capabilities.

According to SDP [RFC4566], the session information ("i=") line has the following syntax:

"i="text

where "text" represents a human-readable text indicating the purpose of the session or media stream.

In this document we define a new capability attribute: the Title capability, "icap". This attribute lists session or media titles as capabilities, according to the following definition:

"a=icap:" title-cap-num 1*WSP text

where <title-cap-num> is a unique integer between 1 and 2^31-1 (both included) user to number the unique ordinal identifier of the particular title capability and <text> is a human-readable text that indicates the purpose of the session or media stream it is supposed to characterize.

As an example, one might use:

a=icap:1 Document Camera

to define a title capability number 1 to identify a particular source of a media stream.

The title capability attribute satisfies the general attribute production rules in SDP [RFC4566] according to the following Augmented Backus-Naur Form (ABNF) [RFC5234] syntax:

```
att-field       = "icap"
att-value       = title-cap-num 1*WSP text
                                ; text defined in RFC4566
title-cap-num    = 1*10(DIGIT)   ; DIGIT defined in RFC5234
```

3.1.2.1. Configuration Parameters

The SDP Capability Negotiation Framework [RFC5939] provides for the existence of the "pcfg" and "acfg" attributes. The concept is extended by the SDP media capabilities negotiation [I-D.ietf-mmusic-sdp-media-capabilities] with an "lcfg" attribute that conveys latent configurations.

In this document, we define an <title-config-list> parameter to be used to convey title capabilities in a potential or latent configuration. This parameter is defined as an <extension-config-list> with the following associations:

ext-cap-name = "i"

ext-cap-list = title-cap-list

This leads to the following definition for the title capability parameter:

```
extension-config-list = title-config-list
title-config-list    = ["+"] "i=" title-cap-list
title-cap-list       = title-cap-num *(BAR title-cap-num)
                                ; BAR defined in RFC5939
title-cap-num        = 1*10(DIGIT) ; DIGIT defined in RFC5234
```

Figure 3: Syntax of the title capability parameter in lcfg and pcfg attributes

Each potential capability configuration contains a single title capability attribute number where 'title-cap-num' is the title capability number defined explicitly earlier in this document, and hence must be between 1 and 2^31-1 (both included). The title capability allows the expression of only a single capability in each alternative, since no more than a single title field is permitted per block. Nevertheless, it is still allowed to express alternative potential title configurations separated by a vertical bar ("|").

An endpoint includes a plus sign ("+") in this configuration attribute to mandate support for this extension. An endpoint that receives this attribute prefixed with a plus sign and does not support this extension MUST treat that potential configuration as not valid.

3.1.2.2. Option Tag

At present, it is difficult to envision a scenario in which the "icap" attribute must be supported or the offer must be rejected. In most cases, if the icap attribute or its contents were to be ignored, an offered configuration could still be chosen based on other criteria such as configuration numbering. However, one might imagine an SDP offer that contained English and Spanish potential configurations for an audio stream. The session might be unintelligible if the choice is based on configuration numbering, rather than informed user selection. Based on such considerations, it may well prove useful to announce the ability to use the icap attribute and its contents to select media configurations, or to inform the user about the selected configuration(s). Therefore, we define a new option tag of "icap-v0" that identifies support for the title capability. This option tag SHOULD be added to other existing option tags present in the "csup" and/or "creq" attributes in SDP, according to the procedures defined in the SDP Capability Negotiation Framework [RFC5939]. The discussion above suggests that "icap-v0" will typically appear in a "csup" attribute, but rarely in a "creq" attribute.

3.2. Session Level versus Media Level

The "icap" and "bcap" attributes can appear at the session level and/or at the media level. Endpoints MUST interpret capabilities declared at session level as part of the session level in the resulting SDP for that particular configuration. Endpoints MUST interpret capabilities declared at media level as part of the media level in the resulting SDP for that particular configuration.

If an "icap" or "bcap" capability for the same bwtype is declared at both session and media level, the media level attribute overrides the value of the session level attribute.

To avoid confusion, the <type-attr-num> for each "a=bcap" and "a=icap" line must be unique across all capability attributes of the same type within the entire session description.

3.3. Offer/Answer model extensions

In this section, we define extensions to the offer/answer model defined in SDP Offer/Answer Model [RFC3264] and extended in the SDP Capability Negotiation [RFC5939] to allow for bandwidth and title capabilities to be used with the SDP Capability Negotiation framework.

3.3.1. Generating the Initial Offer

When an endpoint generates an initial offer and wants to use the functionality described in the current document, it first defines appropriate values for the bandwidth and title capability attributes according to rules defined in [RFC4566] for "b=" and "i=" lines. The endpoint then MUST include the respective capability attributes and associated values in the SDP offer. The preferred configurations for each media stream are identified following the media line in a "pcfg" attribute. Bandwidth and title capabilities may also be referenced in latent configurations, defined in [I-D.ietf-mmusic-sdp-media-capabilities].

The offer SHOULD include the level of capability negotiation extensions needed to support this functionality in a "creq" attribute.

3.3.2. Generating the Answer

When the answering party receives the offer, and if it supports the required capability negotiation extensions, it SHOULD select the most preferred configuration it can support for each media stream, and build the answer accordingly, as defined in Section 3.6.2 of the SDP Capability Negotiation [RFC5939].

3.3.3. Offerer Processing of the Answer

When the offerer receives the answer, it MUST process the media lines according to normal SDP processing rules to identify the media stream(s) accepted by the answer, if any. The "acfg" attribute, if present, may be used to verify the proposed configuration used to form the answer, and to infer the lack of acceptability of higher-preference configurations that were not chosen.

3.3.4. Modifying the Session

If, at a later time, one of the parties wishes to modify the operating parameters of a session, e.g. by adding a new media stream, or by changing the properties used on an existing stream, it may do so via the mechanisms defined for SDP offer/answer [RFC3264].

4. Field Replacement Rules

To simplify the construction of SDP records, given the need to include fields within the media description in question for endpoints that do not support capabilities negotiation, we define some simple field-replacement rules for those fields invoked by potential or

latent configurations. In particular, any "i=" line invoked by a configuration MUST replace the corresponding line, if present within the media description in question. Any "b=" line invoked by a configuration MUST replace any "b=" of the same bandwidth type at the media level.

5. IANA Considerations

5.1. New SDP Attributes

IANA is hereby requested to register new attributes in the "att-field (both session and media level)" of the "Session Description Protocol (SDP) Parameteres" registry, according to the following registration form:

Attribute name: icap

Long form name: Title Capability

Type of attribute: Both media and session level

Subject to charset: Yes

Purpose: Negotiate human-readable information describing the session or media

Appropriate values: See Section 3.1.2 of RFC XXXX

[Note to the RFC Editor: Please replace the above RFC XXXX with the RFC number of this specification.

Contact name: Miguel A. Garcia, Miguel.A.Garcia@ericsson.com

Attribute name: bcap

Long form name: Bandwidth Capability

Type of attribute: Both media and session level

Subject to charset: No

Purpose: Negotiate session or media-level bandwidths

Appropriate values: See Section 3.1.1 of RFC XXXX

[Note to the RFC Editor: Please replace the above RFC XXXX with the RFC number of this specification.

Contact name: Miguel A. Garcia, Miguel.A.Garcia@ericsson.com

5.2. New Option Tags

IANA is hereby requested to add the new option tags "bcap-v0" and "icap-v0", defined herein, to the "SDP Capability Negotiation Option Tag subregistry" of the "Session Description Protocol (SDP) Parameters" registry.

5.3. New SDP Capability Negotiation Configuration Parameters

IANA is hereby requested to add the new parameter identifiers "i" for "title" and "b" for "bandwidth" to the "SDP Capability Negotiation Potential Configuration Parameters" subregistry of the "Session Description Protocol (SDP) Parameters" registry. These parameters are permitted in 'lcfg', 'acfg', and 'pcfg' attributes.

6. Security Considerations

This document provides an extension on top of RFC 4566 [RFC4566], RFC 3264 [RFC3264], SDP Capability Negotiation Framework [RFC5939], and SDP media capabilities negotiation [I-D.ietf-mmusic-sdp-media-capabilities]. As such, the security considerations of those documents apply.

The bandwidth capability attribute may be used for reserving resources at endpoints and intermediaries which inspect the SDP. Modification of the bandwidth value by an attacker can lead to the network being underutilized (too high bandwidth value) or congested (too low bandwidth value). In case it is essential to protect the bandwidth value, one of the security mechanisms proposed in [RFC5939] should be used.

The "i=" line and thus the value carried in the title capability attribute is intended for human-readable description only. It should not be parsed programmatically.

7. Acknowledgments

Thanks to Christer Holmberg, Alf Heidermark, and Ingemar Johansson for arguing for the existence of this document and early reviewing it. Thanks to Flemming Andreasen for a detailed review and many improvement suggestions.

8. References

8.1. Normative References

[I-D.ietf-mmusic-sdp-media-capabilities]
Gilman, R., Even, R., and F. Andreasen, "SDP Media Mapabilities Negotiation", draft-ietf-mmusic-sdp-media-capabilities-11 (work in progress), February 2011.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC5939] Andreasen, F., "Session Description Protocol (SDP) Capability Negotiation", RFC 5939, September 2010.

8.2. Informative References

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[RFC4574] Levin, O. and G. Camarillo, "The Session Description Protocol (SDP) Label Attribute", RFC 4574, August 2006.

Authors' Addresses

Miguel A. Garcia-Martin
Ericsson
Calle Via de los Poblados 13
Madrid, 28033
Spain

Phone: +34 91 339 1000
Email: miguel.a.garcia@ericsson.com

Simo Veikkolainen
Nokia
P.O. Box 407
NOKIA GROUP, FI 00045
Finland

Phone: +358 50 486 4463
Email: simo.veikkolainen@nokia.com

Robert R. Gilman
3243 W. 11th Ave. Dr.
Broomfield, Colorado 80020
U.S.A.

Phone: +1 303 898 9780
Email: bob_gilman@comcast.net

Network Working Group
Internet-Draft
Updates: 5245 (if approved)
Intended status: Standards Track
Expires: August 6, 2011

M. Petit-Huguenin
Stonyfish, Inc.

February 2, 2011

Media level ice-options SDP attribute
draft-petithuguenin-mmusic-ice-attributes-level-00

Abstract

This document redefines the ice-options SDP attribute as a session-level and media-level attribute.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79. This document may not be modified, and derivative works of it may not be created, except to format it for publication as an RFC or to translate it into languages other than English.

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 August 6, 2011.

Copyright Notice

Table of Contents

1. Introduction

ICE [RFC5245] defines the ice-options SDP attribute as session-level only attribute, but when ICE is used with disaggregated media (see section 3 of [I-D.loreto-splices-disaggregated-media]), there is a possibility that different media uses different ICE implementations and/or different networks, and so that differents media in the same SDP require different values for this attribute.

As an example, the ice-options attribute value "rtp+ecn" (defined in [I-D.ietf-avtcore-ecn-for-rtp]) signals ECN capability. Two aggregated media using two different RTP implementations may want to use different values for this attribute.

Note that there is a similar problem for the ice-lite attribute but unfortunately it does not seem possible to design a way to use the ice-lite attribute at the media level that is compatible with legacy implementations that recognize only the session-level attribute.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. ice-options Attribute

The ice-options attribute is a session-level and media-level attribute.

All future new ICE options must also defines how media-level ICE options using this new value are aggregated to eventually generate the value of the session-level ICE option, so legacy implementations that only recognize session-level ICE options can interoperate with implementations that recognize ICE options at both levels.

4. rtp+ecn ICE option

If all aggregated media contains the "rtp+ecn" ICE option defined by [I-D.ietf-avtcore-ecn-for-rtp], then an "rtp+ecn" ICE option MUST be inserted at the session-level.

5. Security Considerations

TBD

6. IANA Considerations

TBD

7. Acknowledgements

This document was written with the xml2rfc tool described in [RFC2629].

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.

[I-D.ietf-avtcore-ecn-for-rtp] Westerlund, M., Johansson, I., Perkins, C., and K. Carlberg, "Explicit Congestion Notification (ECN) for RTP over UDP", draft-ietf-avtcore-ecn-for-rtp-00 (work in progress), January 2011.

8.2. Informative References

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[I-D.loreto-splices-disaggregated-media] Camarillo, G. and S. Loreto, "Disaggregated Media in the Session Initiation Protocol (SIP)", draft-loreto-splices-disaggregated-media-00 (work in progress), September 2010.

Appendix A. Release notes

This section must be removed before publication as an RFC.

Author's Address

Marc Petit-Huguenin
Stonyfish, Inc.

Email: petithug@acm.org

Network WG
Internet-Draft
Expires: September 14, 2011
Intended Status: Standards Track (PS)

James Polk
Subha Dhesikan
Cisco Systems
March 14, 2011

The Session Description Protocol (SDP) 'trafficclass' Attribute
draft-polk-mmusic-traffic-class-for-sdp-01

Abstract

This document proposes a new Session Description Protocol (SDP) attribute to identify the traffic class a session is requesting in its offer/answer exchange.

Table of Contents

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

1. Introduction

The Session Description Protocol (SDP) [RFC4566] provides a means for an offerer to describe the specifics of a session to an answerer, and for the answerer to respond back with its specifics to the offerer. These session specifics include offering the codec or codecs to choose from, the specific IP address and port number the offerer wants to receive the RTP stream(s) on/at, the particulars about the codecs the offerer wants considered or mandated, and so on.

There are many facets within SDP to determine the Real-time

Transport Protocol (RTP) [RFC3550] details established between one or more endpoints, but identifying how the underlying network should process each stream still remains under-specified.

The ability to identify a traffic flow by port number gives an indicate to underlying network elements treat traffic with different ports differently, the same or in groups the same - but different from other ports or groups of ports.

Within the context of realtime communications, the labeling of an RTP session based on media descriptor lines as just a voice and/or video session is insufficient, and provides no guidelines to the underlying network on how to treat the traffic. A more granular labeling helps on several fronts to

- inform application layer elements in the signaling path the intent of this session.

- inform the network on how to treat the traffic if the network is configured to differentiate session treatments based on the type of session the RTP is, including the ability to provide call admission control based on the type of traffic in the network.

- allow network monitoring/management of traffic types realtime and after-the-fact analysis.

Some network operators want the ability to guarantee certain traffic gets a minimum amount of network bandwidth per link or through a series of links that perhaps makes up a network such as a campus or WAN, or a backbone. For example, a call center voice application gets at least 20% of a link as a minimum bandwidth.

Some network operators want the ability to allow certain users or devices access to greater bandwidth during non-busy hours, than during busy hours of the day. For example, all desktop video to operate at 1080p during non-peak times, but curtail a similar session between the same users or devices to 720p or 360p during peak hours. This case is not as clear as accepting or denying similar sessions during different times of the day, but tuning the access to the bandwidth based on the type of session. In other words, tune down the bandwidth for desktop video during peak hours to allow a 3-screen telepresense session that would otherwise look like the same type of traffic (RTP, and more granular, video).

RFC 4594 established a guideline for classifying the various flows in the network and the Differentiated Services Codepoints (DSCP) that apply to many traffic types (table 3 of [RFC4594]), including RTP based voice and video traffic sessions. The RFC also defines the per hop network behavior that is strongly encouraged for each of these application traffic types.

Video was broken down into 4 categories in that RFC, and voice into

another single category. We do not believe this satisfies the technical and business requirements to accomplish sufficiently unique labeling of RTP traffic.

A question arises about once we properly label the traffic, what does that get us? This is a fair question, but out of scope for this document because that answer lies within other RFCs and IDs in other WGs and/or Areas (specifically the Transport Area). That said, we can discuss some of the ideas here for completeness.

If the application becomes aware of traffic labeling,

- this can be coded into layer 3 mechanisms.

- this can be coded into layer 4 protocols and/or mechanisms.

- this can be coded into a combination of mechanisms and protocols.

The layer 3 mechanism for differentiating traffic is either the port number or the Differentiated Services Codepoint (DSCP) value [RFC2474]. Within the public Internet, if the application is not part of a managed service, the DSCP likely will be best effort (BE). Within the corporate LAN, this is usually completely configurable and a local IT department can take full advantage of this labeling to shape and manage their network as they see fit. Communications between enterprise networks will likely have to take advantage of MPLS.

Within a network core, where only MPLS is used, Diffserv typically does not apply. That said, Diffserv can be used to identify which traffic goes into which MPLS tunnels [RFC4124].

Labeling realtime traffic types using a layer 4 protocol would likely mean RSVP [RFC2205] or NSIS [RFC4080]. RSVP has a Application Identifier (app-ID) defined in [RFC2872] that provides a means for carrying a traffic class label along the data path. An advantage with this mechanism is for the label to inform each domain along the media path what type of traffic this traffic flow is, and allow each domain to adjust the appropriate DSCP (set by each domain for use within that domain). Meaning, if a DSCP is set by an endpoint or a router in the first domain and gets reset by a SP, the far end domain will be able to reset the DSCP to the intended traffic class. There is a proposed extension to RSVP which creates individual profiles for what goes into each app-ID field to describe these traffic classes [ID-RSVP-PROF], which will take advantage of what is described in this document.

There are several proprietary mechanisms to take advantage of this labeling, but none of those will be discussed here.

The idea of traffic - or service - identification is not new; it has been described in [RFC5897]. If that RFC is used as a guideline,

identification that leads to stream differentiation can be quite useful. One of the points within RFC 5897 is that users cannot be allowed to assign any identification (fraud is but one reason given). In addition, RFC 5897 recommends that service identification should be done in signaling, rather than guessing or deep packet inspection. The network will have to currently guess or perform deep packet inspection to classify and offer the service as per RFC 4594 since such service identification information is currently not available in SDP and therefore to the network elements. Since SDP understands how each stream is created (i.e., the particulars of the RTP stream), this is the right place to have this service differentiated. Such service differentiation can then be communicated to and leveraged by the network.

[Editor's Note: the words "traffic" and "service" are similar enough that the above paragraph talks about RFC 5897's "service identification", but this document is only wanting to discuss and propose traffic indications in SDP.]

This document proposes a simple attribute line to identify the application a session is requesting in its offer/answer exchange. This document uses previously defined service class strings for consistency between IETF documents.

2. SDP Attribute Definition

This document proposes the 'trafficclass' session and media-level SDP [RFC4566] attribute. The following is the Augmented Backus-Naur Form (ABNF) [RFC5234] syntax for this attribute, which is based on the SDP [RFC4566] grammar:

```
attribute               =/ traffic-classification

traffic-classification  = "trafficclass" ":" [SP] app-type
                            *( add-param )

app-type                = "Broadcast-video" /
                          "Realtime-Interactive" /
                          "Multimedia-Conferencing" /
                          "Multimedia-Streaming" /
                          "Telephony" /  "Voice-Admit" /
                          "unknown" / extension-mech

extension-mech          = token

add-param               = "." sub-app-type / "." cac-class

sub-app-type            = "telepresence" / "immersive" /
                          "desktop" / "personal" /"webex" /
                          "call center" / "trading floor" /
```

```
                              "handheld" / generic-param ; from
                               RFC3261

   cac-class               = "admitted" / "non-admitted"
```

The attribute is named "trafficclass", for traffic classification, identifying which one of the six traffic classes applies to the media stream. There MUST NOT be more than one trafficclass attribute per media line. Confusion would result in where more than one exists per m= line.

The application type traffic classes defined in this document for SDP are an augmented version of the application labels introduced by table 3 of RFC 4594. RFC 5685 updated the guidelines set forth in RFC 4594 by creating a new voice classification where call admission control (CAC) has been applied. There are four video classifications and two voice classifications

- Broadcast-video
- Realtime-Interactive
- Multimedia-Conferencing
- Multimedia-Streaming
- Telephony
- Voice-Admit
- unknown

The "unknown" application type is for the scenario in which the application type is not known, but the sub-application type is.

The application types (app-type) can be further subdivided into sub-application types with the sub-app-type identifiers for more granular application distinction of the traffic. Sub-application types are separated from traffic class by a "." if any are present in an instance of this attribute. One or more sub-app-types MAY be present in the trafficclass attribute. There MUST NOT be more than one application type in a single instance of the trafficclass attribute. If there is a sub-application type, there MUST be an application type, where the "unknown" is permissible.

This document creates the following sub-application types

- telepresence
- immersive
- desktop
- personal
- webex
- call center
- trading floor
- handheld

In addition to, of as an alternative to one or more sub-application types, a cac-class value MAY be present indicating whether or not a

session has had call admission control applied to it. The following two values are created by this document for the cac-class value:

- admitted
- nonadmitted

The default cac-class value for any trafficclass attribute is nonadmitted, even if not present.

Any application, sub-application or cac-class not understood MUST be ignored, leaving all that is understood to be processed.

The following is an example of media level description with a 'trafficclass' attribute:

```
m=audio 50000 RTP/AVP 112
a=trafficclass multimedia-conferencing.telepresence.immersive.
               admitted
```

The above indicates a multiscreen telepresence session that has had call admission control applied to the traffic.

An sub-application type does not have to be defined within this document or an update/extension to this document to be used. The 'trafficclass' attribute is allowed to have one or more vendor specific (i.e., proprietary) sub-application types. These vendor specific sub-application types MUST have an underscore "_" character immediately after one of the "." characters in the 'trafficclass' attribute.

The following is an example of media level description with a 'trafficclass' attribute that has proprietary sub-application identifiers:

```
m=audio 50000 RTP/AVP 0
a=trafficclass multimedia-conferencing.telepresence._foo._bar
```

3.0 Offer/Answer Behavior

Through the inclusion of the 'trafficclass' attribute, an offer/answer exchange identifies the application type for use by endpoints within a session. Policy elements can use this attribute to determine the acceptability and/or treatment of that session through lower layers. One specific use-case is for setting of the DSCP specific for that application type (say Broadcast Video instead of Real-time Interactive video), decided on a per domain basis - instead of exclusively by the offering domain.

3.1 Offer Behavior

Offerers include the 'trafficclass' attribute with a single well known token (from list in Section 2) to obtain configurable and predictable treatment between the answerer and the offerer. It can also instead include a private token within a single domain (e.g., enterprise networks).

Offerers of this 'trafficclass' attribute MUST NOT change the token in transit (e.g., wrt to B2BUAs). SBCs at domain boundaries can change this attribute through local policy.

Offers containing a 'trafficclass' token not understood are ignored by default (i.e., as if there was no 'trafficclass' attribute in the Offer).

3.2 Answer Behavior

Upon receiving an offer containing a 'trafficclass' attribute, if the offer is accepted, the answerer will use this attribute to set the session or media (level) traffic accordingly towards the offerer.

The answerer will answer the offer with its own 'trafficclass' attribute, which will likely be the same value, although this is not mandatory (at this time).

The answerer should expect to receive RTP packets marked as indicated by its 'trafficclass' attribute in the answer itself.

An Answer MAY have a 'trafficclass' attribute when one was not in the offer. This will at least aid the local domain, and perhaps each domain the session transits, to categorize the application type of this RTP session.

Answerers that are middleboxes can use the 'trafficclass' attribute to classify the RTP traffic within this session however local policy determines. In other words, this attribute can help in deciding which DSCP an RTP stream is assigned within a domain, if the answerer were an inbound SBC to a domain.

4. Security considerations

RFC 5897 [RFC5897] discusses many of the pitfalls of service classification, which is similar enough to this idea of traffic classification to apply here as well. That document highly recommends the user not being able to set any classification. Barring a hack within an endpoint (i.e., to intentionally mis-classifying (i.e., lying) about which classification an RTP stream is), this document's solution makes the classification part

of the signaling between endpoints, which is recommended by RFC 5897.

5. IANA considerations

5.1 Registration of the SDP 'trafficclass' Attribute

This document requests IANA to register the following SDP att-field under the Session Description Protocol (SDP) Parameters registry:

Contact name: jmpolk@cisco.com

Attribute name: trafficclass

Long-form attribute name: Traffic Classification

Type of attribute: Session and Media levels

Subject to charset: No

Purpose of attribute: To indicate the Traffic Classification application for this session

Allowed attribute values: IANA Registered Tokens

Registration Procedures: Specification Required

Type	SDP Name	Reference
att-field (both session and media level)		
	trafficclass	[this document]

5.2 The Traffic Classification Application Type Registration

This document requests IANA to create a new registry for the traffic application classes similar to the following table within the Session Description Protocol (SDP) Parameters registry:

Registry Name: "trafficclass" SDP Application Type Attribute Values
Reference: [this document]
Registration Procedures: Specification Required

Attribute Values	Reference
Broadcast video	[this document]
Real-time Interactive	[this document]
Multimedia Conferencing	[this document]
Multimedia Streaming	[this document]
Telephony	[this document]
Voice-Admit	[this document]

5.3 The Traffic Classification Sub-Application Type Registration

This document requests IANA to create a new registry for the traffic sub-application classes similar to the following table within the Session Description Protocol (SDP) Parameters registry:

Registry Name: "trafficclass" Attribute Sub-Application Type Values
Reference: [this document]
Registration Procedures: Specification Required

Attribute Values	Reference
Telepresence	[this document]
immersive	[this document]
desktop	[this document]
personal	[this document]
webex	[this document]
call center	[this document]
trading floor	[this document]
handheld	[this document]

5.4 The Traffic Classification Attribute Call Admission Control Class Registration

This document requests IANA to create a new registry for the Call Admission Control Class similar to the following table within the Session Description Protocol (SDP) Parameters registry:

Registry Name: "trafficclass" SDP Call Admission Control Class (cac-class) Attribute Values
Reference: [this document]
Registration Procedures: Specification Required

Attribute Values	Reference
Admitted	[this document]
Non-admitted	[this document]

6. Acknowledgments

To Dave Oran, Toerless Eckert, Henry Chen, David Benham and Paul Jones for their comments and suggestions.

7. References

7.1. Normative References

[RFC2119] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997

[RFC4566] M. Handley, V. Jacobson, C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC5865] F. Baker, J. Polk, M. Dolly, "A Differentiated Services Code Point (DSCP) for Capacity-Admitted Traffic", RFC 5865, May 2010

[RFC2474] K. Nichols, S. Blake, F. Baker, D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers ", RFC 2474, December 1998

[RFC2205] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997

[RFC4080] R. Hancock, G. Karagiannis, J. Loughney, S. Van den Bosch, "Next Steps in Signaling (NSIS): Framework", RFC 4080, June 2005

[RFC2872] Y. Bernet, R. Pabbati, "Application and Sub Application Identity Policy Element for Use with RSVP", RFC 2872, June 2000

[RFC5897] J. Rosenberg, "Identification of Communications Services in the Session Initiation Protocol (SIP)", RFC 5897, June 2010

[RFC4124] F. Le Faucheur, Ed., " Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering ", RFC 4124, June 2005

[RFC5234] Crocker, D., Ed., and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

7.2. Informative References

[RFC4594] J. Babiarz, K. Chan, F Baker, "Configuration Guidelines for Diffserv Service Classes", RFC 4594, August 2006

[ID-RSVP-PROF] J. Polk, S. Dhesikan, "Resource Reservation Protocol (RSVP) Application-ID Profiles for Voice and Video Streams", work in progress, Mar 2011

Author’s Addresses

James Polk
3913 Treemont Circle
Colleyville, Texas, USA
+1.817.271.3552

mailto: jmpolk@cisco.com

Subha Dhesikan
170 W Tasman St
San Jose, CA, USA
+1.408-902-3351

mailto: sdhesika@cisco.com