

MMUSIC Working Group
Internet-Draft
Intended status: Experimental
Expires: January 10, 2013

V. Singh
J. Ott
T. Karkkainen
Aalto University
R. Globisch
T. Schierl
Fraunhofer HHI
July 9, 2012

**Multipath RTP (MPRTP) attribute in Session Description Protocol
draft-singh-mmusic-mprtp-sdp-extension-00**

Abstract

Multipath RTP (MPRTP) is an extension to the Real-time Transport Protocol (RTP) that allows multi-homed endpoints to take advantage of the availability of multiple Internet paths between endpoints to send/receive media packets. This document describes how to express the interface advertisement and negotiation during session setup in SDP (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 January 10, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

| | | |
|-----------------------------|--|--------------------|
| 1. | Introduction | 4 |
| 1.1. | Requirements Language | 4 |
| 1.2. | Terminology | 4 |
| 2. | SDP Considerations | 4 |
| 2.1. | MPRTP Interface Advertisement in SDP (out-of-band signaling) | 4 |
| 2.1.1. | "interface" attribute | 5 |
| 2.1.2. | Example | 6 |
| 2.2. | MPRTP with ICE | 6 |
| 2.3. | Offer/Answer | 7 |
| 2.3.1. | In-band Signaling Example | 7 |
| 2.3.2. | Out-of-band Signaling Example | 8 |
| 2.3.2.1. | Without ICE | 8 |
| 2.3.2.2. | With ICE | 9 |
| 2.4. | Increased Throughput | 11 |
| 2.5. | Increased Reliability | 11 |
| 2.6. | Decoding dependency | 11 |
| 3. | MPRTP in RTSP | 11 |
| 3.1. | Solution Overview without ICE | 11 |
| 3.2. | Solution Overview with ICE | 13 |
| 3.3. | RTSP Extensions | 15 |
| 3.3.1. | MPRTP Interface Transport Header Parameter | 15 |
| 3.3.2. | MPRTP Feature Tag | 16 |
| 3.3.3. | Status Codes | 16 |
| 3.3.4. | New Reason for PLAY_NOTIFY | 16 |
| 3.3.5. | Re-SETUP | 17 |
| 4. | IANA Considerations | 17 |
| 4.1. | SDP Attributes | 18 |
| 4.1.1. | "mprtp" attribute | 18 |
| 4.2. | RTSP | 18 |
| 4.2.1. | RTSP Feature Tag | 18 |
| 4.2.2. | RTSP Transport Parameters | 18 |
| 4.2.3. | Notify-Reason value | 18 |
| 5. | Security Considerations | 19 |
| 6. | Acknowledgements | 19 |
| 7. | Contributors | 19 |
| 8. | References | 20 |
| 8.1. | Normative References | 20 |
| 8.2. | Informative References | 20 |
| Appendix A. | Change Log | 21 |
| A.1. | Changes in draft-singh-mmusic-mprtp-sdp-extension-00 | 21 |
| | Authors' Addresses | 21 |

1. Introduction

Multipath RTP (MPRTP) [1] is an extension to RTP [2] that allows splitting a single RTP stream into multiple subflows, which are then transmitted over different Internet paths. Multipath RTCP (MPRTCP) [1] is an extension to RTCP. It is used along with MPRTP to report per-path sender and receiver characteristics.

A Multipath RTP session can be set up in many possible ways e.g., during handshake, or upgraded mid-session. The capability exchange may be done using out-of-band signaling (e.g., Session Description Protocol (SDP) [7] in Session Initiation Protocol (SIP) [8], Real-Time Streaming Protocol (RTSP) [9]) or using in-band signaling (e.g., in RTCP [1]).

This document defines an extension to the SDP attribute 'a=mprtp' defined in the base MPRTP specification [1]. Using this extension an endpoint can advertise its multiple interfaces.

1.1. Requirements Language

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 [3].

1.2. Terminology

The definitions for the words Endpoint, Interface, Path and Subflow in this document are as per [1].

2. SDP Considerations

The base Multipath RTP specification [1] defines the 'a=mprtp' attribute to indicate support for MPRTP to its peers. In the following section, we extend the 'a=mprtp' attribute to advertise an endpoint's multiple interfaces in SDP instead of advertising the interfaces in-band in RTCP [1].

2.1. MPRTP Interface Advertisement in SDP (out-of-band signaling)

If the endpoint is aware of its multiple interfaces and wants to use them for MPRTP, it MAY use SDP to advertise these interfaces. Alternatively, it MAY use in-band signaling to advertise its interfaces, as defined in [1]. The receiving endpoint MUST use the same mechanism to respond to an interface advertisement. In particular, if an endpoint receives an SDP containing multiple MPRTP interfaces, then it MUST respond to the offer in SDP with its set of

MPRTP interfaces.

2.1.1. "interface" attribute

The interface attribute is an optional media-level attribute and is used to advertise an endpoint's interface address.

The syntax of the interface attribute is defined using the following Augmented BNF, as defined in [4]. The definitions of unicast-address, port, token, SP, and CRLF are according to RFC4566 [10].

```
mprtp-optional-parameter = mprtp-interface
                           ; other optional parameters may be added later

mprtp-interface = "interface" ":" counter SP unicast-address
                  ":" rtp_port
                  *(SP interface-description-extension)

counter = 1*DIGIT
rtp_port = port      ;port from RFC4566
```

<mprtp-interface>: specifies one unicast IP address, the RTP and RTCP port number of the endpoint. The unicast address with lowest counter value MUST match the connection address ('c=' line). Similarly, the RTP and RTCP ports MUST match the RTP and RTCP ports in the associated 'm=' line. The counter SHOULD start at 1 and increment with each additional interface. Multiple interface lines MUST be ordered in a decreasing priority level as is the case with the Interface Advertisement blocks in in-band signaling (See [1]).

<unicast-address>: is taken from RFC4566 [10]. It is one of the IP addresses of the endpoint and allows the use of IPv4 addresses, IPv6 addresses and Fully Qualified Domain Names (FQDN). An endpoint MUST only include the IP address for which the connectivity checks have succeeded.

<port>: is from RFC4566 [10]. It is the RTP port associated with the unicast address and note that the RTP and RTCP ports are multiplexed for MPRTP subflows.

<counter>: is a monotonically increasing positive integer starting at 1. The counter MUST reset for each media line. The counter value for an 'mprtp-interface' should remain the same for the session.

The 'mprtp-interface' can be extended using the 'interface-description-extension' parameter. An endpoint MUST ignore any extensions it does not understand.

[2.1.2.](#) Example

The ABNF grammar is illustrated by means of an example:

```
v=0
o=alice 2890844526 2890844527 IN IP4 192.0.2.1
s=
c=IN IP4 192.0.2.1
t=0 0
m=video 49170 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E;
a=extmap:1 urn:ietf:params:rtp-hdext:mprtp
a=rtcp-mux
a=mprtp interface:1 192.0.2.1:49170      ;primary interface
a=mprtp interface:2 198.51.100.1:51372   ;additional interface
```

[2.2.](#) MPRTTP with ICE

If the endpoints intend to use ICE [\[5\]](#) for discovering interfaces and running connectivity checks, the following two step procedure MUST be followed:

1. Advertise ICE candidates: in the initial OFFER the endpoints exchange candidates, as defined in ICE [\[5\]](#). Thereafter the endpoints run connectivity checks.
2. Advertise MPRTTP interfaces: When a sufficient number of connectivity checks succeed, the endpoint MUST send an updated offer containing the interfaces that they want to use for MPRTTP.

When an endpoint uses ICE's regular nomination [\[5\]](#) procedure, it chooses the best ICE candidate as the default path. In the case of an MPRTTP endpoint, if more than one ICE candidate succeeded the connectivity checks then an MPRTTP endpoint MAY advertise (some of) these as MPRTTP interfaces in an updated offer.

When an endpoint uses ICE's aggressive nomination [\[5\]](#) procedure, the selected candidate may change as more ICE checks complete. Instead of sending updated offers as additional ICE candidates appear (transience), the endpoint MAY use in-band signaling to advertise its interfaces, as defined in [\[1\]](#). Additionally, it MAY send an updated offer when the transience stabilizes.

If the default interface disappears and the paths used for MPRTTP are different from the one in the c= and m= lines then the 'mprtp interface' with the lowest counter value should be promoted to the c= and m= lines in the updated offer.

When a new interface appears, then the application/endpoint should internally decide if it wishes to use it and sends an updated offer with ICE candidates of the new interface. The receiving endpoint responds to the offer with all its ICE candidates in the answer and starts connectivity checks between all its candidates and the offerer's new ICE candidate. Similarly, the initiating endpoint starts connectivity checks between the new interface and all the received ICE candidates in the answer. If the connectivity checks succeed, the initiating endpoint MAY send an updated offer with the new interface as an additional 'mptp interface'.

2.3. Offer/Answer

When SDP [10] is used to negotiate MP RTP interfaces (see [Section 2.1](#)) following the offer/answer model [7], the collection of "a=mptp interface" attribute lines indicates the interfaces the endpoint wishes to use for sending and/or receiving media data. The SDP offer MUST include this attribute at the media level. If the answerer wishes to also use SDP for advertising MP RTP interfaces, it MUST also include its interfaces at the media-level "a=mptp interface" attribute in the answer. If the answer does not contain an "a=mptp interface" attribute, the offerer MUST use in-band signaling [1] for advertising interfaces.

When SDP is used in a declarative manner, the presence of an "a=mptp interface" attribute signals that the sender can send or receive media data over multiple interfaces. The receiver SHOULD be capable to stream media to the multiple interfaces and be prepared to receive media from multiple interfaces.

The following sections shows examples of SDP offer and answer for in-band and out-of-band signaling.

2.3.1. In-band Signaling Example

The following offer/answer shows that both the endpoints are MP RTP capable and SHOULD use in-band signaling for interfaces advertisements.

Offer:

```
v=0
o=alice 2890844526 2890844527 IN IP4 192.0.2.1
s=
c=IN IP4 192.0.2.1
t=0 0
m=video 49170 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E;
a=rtcp-mux
a=mprtp
```

Answer:

```
v=0
o=bob 2890844528 2890844529 IN IP4 192.0.2.2
s=
c=IN IP4 192.0.2.2
t=0 0
m=video 4000 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E;
a=rtcp-mux
a=mprtp
```

The endpoint MAY now use in-band RTCP signaling to advertise its multiple interfaces. Alternatively, it MAY make another offer with the interfaces in SDP (out-of-band signaling).

2.3.2. Out-of-band Signaling Example

If the multiple interfaces are included in an SDP offer then the receiver MUST respond to the request with an SDP answer.

2.3.2.1. Without ICE

In this example, the offerer advertises two interfaces and the answerer responds with a single interface description. The endpoint MAY use one or both paths depending on the end-to-end characteristics of each path.

Offer:

```
v=0
o=alice 2890844526 2890844527 IN IP4 192.0.2.1
s=
c=IN IP4 192.0.2.1
t=0 0
m=video 49170 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E;
a=rtcp-mux
a=mp RTP interface:1 192.0.2.1:49170
a=mp RTP interface:2 198.51.100.1:51372
```

Answer:

```
v=0
o=bob 2890844528 2890844529 IN IP4 192.0.2.2
s=
c=IN IP4 192.0.2.2
t=0 0
m=video 4000 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E;
a=rtcp-mux
a=mp RTP interface:1 192.0.2.2:4000
```

[2.3.2.2.](#) With ICE

In this example, the endpoint first sends its ICE candidates in the initial offer and the other endpoint answers with its ICE candidates.

Initial offer (with ICE candidates):

Offer:

```
v=0
o=alice 2890844526 2890844527 IN IP4 192.0.2.1
s=
c=IN IP4 192.0.2.1
t=0 0
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
a=mp RTP
m=video 49170 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E;
a=rtcp-mux
a=candidate:1 1 UDP 2130706431 192.0.2.1 49170 typ host
a=candidate:2 1 UDP 1694498815 198.51.100.1 51372 typ host
```


Answer:

```
v=0
o=bob 2890844528 2890844529 IN IP4 192.0.2.2
s=
c=IN IP4 192.0.2.2
t=0 0
a=ice-pwd:YH75Fviy6338Vbrhrlp8Yh
a=ice-ufrag:9uB6
a=mrtp
m=video 4000 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E;
a=rtcp-mux
a=candidate:1 1 UDP 2130706431 192.0.2.2 4000 typ host
```

Thereafter, each endpoint conducts ICE connectivity checks and when sufficient number of connectivity checks succeed, the endpoint sends an updated offer. In the updated offer, the endpoint advertises its multiple interfaces for MPRTTP.

Updated offer (with MPRTTP interfaces):

Offer:

```
v=0
o=alice 2890844526 2890844527 IN IP4 192.0.2.1
s=
c=IN IP4 192.0.2.1
t=0 0
m=video 49170 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E;
a=rtcp-mux
a=candidate:1 1 UDP 2130706431 192.0.2.1 49170 typ host
a=candidate:2 1 UDP 1694498815 198.51.100.1 51372 typ host
a=mrtp interface:1 192.0.2.1:49170
a=mrtp interface:2 198.51.100.1:51372
```


Answer:

```
v=0
o=bob 2890844528 2890844529 IN IP4 192.0.2.2
s=
c=IN IP4 192.0.2.2
t=0 0
m=video 4000 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E;
a=rtcp-mux
a=candidate:1 1 UDP 2130706431 192.0.2.2 4000 typ host
a=mp RTP interface:1 192.0.2.2:4000
```

2.4. Increased Throughput

The MP RTP layer MAY choose to augment paths to increase throughput. If the desired media rate exceeds the current media rate, the endpoints MUST renegotiate the application specific ("b=AS:xxx") [[10](#)] bandwidth.

2.5. Increased Reliability

TBD

2.6. Decoding dependency

TBD

3. MP RTP in RTSP

Endpoints MUST use RTSP 2.0 [[9](#)] for session setup. Endpoints MUST multiplex RTP and RTCP on a single port [[6](#)] and follow the recommendations made in [Appendix C](#) of [[9](#)].

3.1. Solution Overview without ICE

1. The RTSP Server should include all of its interfaces via the SDP attribute ("a=mp RTP interface") in the RTSP DESCRIBE message.
2. The RTSP Client should include its multiple interface in the RTSP SETUP message using the new attribute ("dest_mp RTP_addr=") in the Transport header.
3. The RTSP Server responds to the RTSP SETUP message with a 200 OK containing its MP RTP interfaces (using the "src_mp RTP_header=") in the Transport header. After this, the RTSP Client can issue a PLAY request.

4. If a new interface appears or an old one disappears at the RTSP Client during playback, it should send a new RTSP SETUP message containing the updated interfaces ("dest_mprtp_addr") in the Transport header.
5. If a new interface appears or an old one disappears at the RTSP Server during playback, the RTSP Server should send a PLAY_NOTIFY message with a new Notify-Reason: "src-mprtp-interface-update". The request must contain the updated interfaces ("dest_mprtp_addr") in the "MPRTP-Interfaces" header.
6. Alternatively, the RTSP Server or Client may use the RTCP (in-band) mechanism to advertise their interfaces.

The overview is illustrated by means of an example:

```
C->S: DESCRIBE rtsp://server.example.com/fizzle/foo RTSP/2.0
      CSeq: 111
      User-Agent: PhonyClient 1.3
      Accept: application/sdp, application/example
      Supported: setup.mprtp, setup.rtp.rtcp.mux

S->C: RTSP/2.0 200 OK
      CSeq: 111
      Date: 23 Jan 2011 15:35:06 GMT
      Server: PhonyServer 1.3
      Content-Type: application/sdp
      Content-Length: 367
      Supported: setup.mprtp, setup.rtp.rtcp.mux

v=0
o=mprtp-rtsp-server 2890844526 2890844527 IN IP4 192.0.2.1
s=
c=IN IP4 192.0.2.1
t=0 0
m=video 49170 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E;
a=extmap:1 urn:ietf:params:rtp-hdext:mprtp
a=rtcp-mux
a=mprtp interface:1 192.0.2.1:49170
a=mprtp interface:2 198.51.100.1:51372
```

On receiving the response to the RTSP DESCRIBE message, the RTSP Client sends an RTSP SETUP message containing its MPRTP interfaces in the Transport header using the "dest_mprtp_addr=" attribute. The RTSP Server responds with a 200 OK containing both the RTSP Client's and the RTSP Server's MPRTP interfaces.


```
C->S: SETUP rtsp://server.example.com/fizzle/foo/audio RTSP/2.0
      CSeq: 112
      Transport: RTP/AVPF/UDP; unicast; dest_mprtp_addr="
1 192.0.2.2 4000"; RTCP-mux,
      RTP/AVP/UDP; unicast; dest_addr=":6970"/":6971",
      RTP/AVP/TCP;unicast;interleaved=0-1
      Accept-Ranges: NPT, UTC
      User-Agent: PhonyClient 1.3
      Supported: setup.mprtp, setup.rtp.rtcp.mux

S->C: RTSP/2.0 200 OK
      CSeq: 112
      Session: 12345678
      Transport: RTP/AVPF/UDP; unicast; dest_mprtp_addr="
1 192.0.2.2 4000";
      src_mprtp_addr="1 192.0.2.1 49170;
2 198.51.100.1 51372"; RTCP-mux
      Accept-Ranges: NPT
      Date: 23 Jan 2012 15:35:06 GMT
      Server: PhonyServer 1.3
      Supported: setup.mprtp, setup.rtp.rtcp.mux
```

The RTSP Client can issue a PLAY request on receiving the 200 OK and media can start to stream once the RTSP Server receives the PLAY request.

3.2. Solution Overview with ICE

This overview uses the ICE mechanisms [[11](#)] defined for RTSP 2.0 [[9](#)].

1. The RTSP Server should include the "a=rtsp-ice-d-m" attribute and also indicate that it supports MPRTTP by including the "a=mprtp" attribute in the SDP of the RTSP DESCRIBE message.
2. The client sends an RTSP SETUP message containing the D-ICE in lower level transport and ICE candidates in the transport header. The RTSP Server and Client then follow the procedures (Steps 2 to 8) described in [[11](#)].
3. When the connectivity checks conclude, the RTSP Client can send an updated RTSP SETUP message with its MPRTTP interfaces (ICE candidates that were successful) in the Transport header ("dest_mprtp_addr="). The RTSP Server responds to the RTSP SETUP message with a 200 OK containing its MPRTTP interfaces (ICE candidates that were successful) in the Transport header ("src_mprtp_header="). After receiving the 200 OK, the RTSP Client can issue the PLAY request.

4. Alternatively, after the connectivity checks conclude, the RTSP Client can issue the PLAY request (Step 9 and 10 of [11]) and the endpoints can use the RTCP (in-band) mechanism to advertise their interfaces.
5. If a new interface appears or an old one disappears, the RTSP Client should issue an updated SETUP message with the new candidates (See Section 5.12 of [11]) or the RTSP Server should send a PLAY_NOTIFY message (See Section 5.13 of [11]). After connectivity checks succeed for the new interfaces, the RTSP Client can proceed with the instructions in Step 3 or 4.

The overview is illustrated by means of an example:

```
C->S: DESCRIBE rtsp://server.example.com/foo RTSP/2.0
      CSeq: 312
      User-Agent: PhonyClient 1.3
      Accept: application/sdp, application/example
      Supported: setup.mprtp, setup.ice-d-m, setup.rtp.rtcp.mux

S->C: RTSP/2.0 200 OK
      CSeq: 312
      Date: 23 Jan 2012 15:35:06 GMT
      Server: PhonyServer 1.3
      Content-Type: application/sdp
      Content-Length: 367
      Supported: setup.mprtp, setup.ice-d-m, setup.rtp.rtcp.mux

v=0
o=mprtp-rtsp-server 2890844526 2890842807 IN IP4 192.0.2.1
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.example.com/lectures/sdp.ps
e=seminar@example.com (Seminar Management)
t=2873397496 2873404696
a=recvonly
a=rtsp-ice-d-m
a=control: *
m=video 49170 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E;
a=rtcp-mux
a=mprtp
a=control: /video
```



```
C->S: SETUP rtsp://server.example.com/foo/video RTSP/2.0
      CSeq: 302
      Transport: RTP/AVP/D-ICE; unicast; ICE-ufrag=9uB6;
                ICE-Password=YH75Fviy6338Vbrhrlp8Yh;
                candidates="1 1 UDP 2130706431 192.0.2.2
                4000 typ host"; RTCP-mux,
                RTP/AVP/UDP; unicast; dest_addr=":6970"/":6971",
                RTP/AVP/TCP;unicast;interleaved=0-1
      Accept-Ranges: NPT, UTC
      User-Agent: PhonyClient 1.3
      Supported: setup.mprtp, setup.ice-d-m, setup.rtp.rtcp.mux

S->C: RTSP/2.0 200 OK
      CSeq: 302
      Session: 12345678
      Transport: RTP/AVP/D-ICE; unicast; RTCP-mux;
                ICE-ufrag=8hhY; ICE-Password=
                asd88fgpdd777uzjYhagZg; candidates="
                1 1 UDP 2130706431 192.0.2.1 49170 typ host;
                2 1 UDP 1694498815 198.51.100.1 51372 typ host"
      Accept-Ranges: NPT
      Date: 23 Jan 2012 15:35:06 GMT
      Server: PhonyServer 1.3
      Supported: setup.mprtp, setup.ice-d-m, setup.rtp.rtcp.mux
```

After the connectivity checks complete, the RTSP Client can send an updated RTSP SETUP message containing the MPRTTP interfaces for which the connectivity checks were successful. These steps are the same as the ones in the previous example.

[3.3. RTSP Extensions](#)

[3.3.1. MPRTTP Interface Transport Header Parameter](#)

This section defines a new RTSP transport parameter for carrying MPRTTP interfaces. The transport parameters may only occur once in each transport specification. The parameter can contain one or more MPRTTP interfaces. In the SETUP response if the RTSP Server supports MPRTTP it MUST include one or more MPRTTP interfaces.


```

trns-parameter = <Defined in Section 20.2.3 of
                    [I-D.ietf-mmusic-rfc2326bis]>
trns-parameter =/ SEMI dest-mprtp-interface-par
trns-parameter =/ SEMI src-mprtp-interface-par
dest-mprtp-interface-par = "dest_mprtp_addr" EQUAL DQ SWS
                        interface *(SEMI interface) SWS DQ
src-mprtp-interface-par = "src_mprtp_addr" EQUAL DQ SWS
                        interface *(SEMI interface) SWS DQ

interface      = counter SP
                unicast-address SP
                rtp_port SP
                *(SP interface-description-extension)

counter        = See section 2.3.1
unicast-address = See section 2.3.1
rtp_port       = See section 2.3.1
interface-description-extension = See section 2.3.1

```

[3.3.2.](#) MP RTP Feature Tag

A feature tag is defined for indicating MP RTP support in the RTSP capabilities mechanism: "setup.mprtp". This feature tag indicates that the endpoint supports all the mandatory extensions defined in this specification and is applicable to all types of RTSP agents; clients, servers and proxies.

The MP RTP compliant RTSP Client MUST send the feature tag "setup.mprtp" in the "Supported" header of all DESCRIBE and SETUP requests.

[3.3.3.](#) Status Codes

TBD

[3.3.4.](#) New Reason for PLAY_NOTIFY

A new value used in the PLAY_NOTIFY methods Notify-Reason header is defined: "src-mprtp-interface-update". This reason indicates that the RTSP Server has updated set of MP RTP interfaces.

```
Notify-Reas-val =/ "src-mprtp-interface-update"
```

PLAY_NOTIFY requests with Notify-Reason header set to src-mprtp-interface-update MUST include a mprtp-interfaces header.


```
mprtp-interfaces      = "mprtp-interfaces" HCOLON interface
                        *(COMMA interface)
interface = counter SP
            unicast-address SP
            rtp_port SP
            *(SP interface-description-extension)

counter      = See section 2.3.1
unicast-address = See section 2.3.1
rtp_port     = See section 2.3.1
interface-description-extension = See section 2.3.1
```

Example:

```
S->C: PLAY_NOTIFY rtsp://server.example.com/foo RTSP/2.0
      CSeq: 305
      Notify-Reason: src-mprtp-interface-update
      Session: 12345678
      mprtp-interfaces: 2 192.0.2.10 48211, 3 198.51.100.11 38703
      Server: PhonyServer 1.3

C->S: RTSP/2.0 200 OK
      CSeq: 305
      User-Agent: PhonyClient 1.3
```

[3.3.5.](#) Re-SETUP

The server SHALL support SETUP requests in PLAYING state if it is only updating the transport parameter (dest_mprtp_addr). If the session is established using ICE then the RTSP Server and Client MUST also follow the procedures described for Re-SETUP in [[11](#)].

[4.](#) IANA Considerations

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

```
Contact:    Varun Singh
            mailto:varun.singh@iki.fi
            tel:+358-9-470-24785
```

Note to the RFC-Editor: When publishing this document as an RFC, please replace "RFC XXXX" with the actual RFC number of this document and delete this sentence.

4.1. SDP Attributes

4.1.1. "mprtp" attribute

- o Attribute Name: MPRTTP
- o Long Form: Multipath RTP
- o Type of Attribute: media-level
- o Charset Considerations: The attribute is not subject to the charset attribute.
- o Purpose: This attribute is extended to signal one of many possible interfaces for communication. These interface addresses may have been validated using ICE procedures.
- o Appropriate Values: [Section 2.1.1](#) of RFC XXXX.

4.2. RTSP

This document requests registration in a number of registries for RTSP.

4.2.1. RTSP Feature Tag

This document request that one RTSP 2.0 feature tag be registered in the "RTSP 2.0 feature tag" registry:

setup.mprtp See [Section 3.3.2](#).

4.2.2. RTSP Transport Parameters

This document requests that 2 transport parameters be registered in RTSP 2.0's "Transport Parameters":

"dest_mprtp_addr": See [Section 3.3.1](#).

"src_mprtp_addr": See [Section 3.3.1](#).

4.2.3. Notify-Reason value

This document requests that one assignment be done in the RTSP 2.0 Notify-Reason header value registry. The defined value is:

"src-mprtp-interface-update": See [Section 3.3.4](#).

5. Security Considerations

All drafts are required to have a security considerations section. See [RFC 3552](#) [[12](#)] for a guide.

6. Acknowledgements

Varun Singh, Saba Ahsan, and Teemu Karkkainen are supported by Trilogy (<http://www.trilogy-project.org>), a research project (ICT-216372) partially funded by the European Community under its Seventh Framework Program. The views expressed here are those of the author(s) only. The European Commission is not liable for any use that may be made of the information in this document.

The authors would also like acknowledge the contribution of Ralf Globisch and Thomas Schierl for providing input and text for the MP RTP interface advertisement in SDP.

7. Contributors

Saba Ahsan
Aalto University
School of Science and Technology
Otakaari 5 A
Espoo, FIN 02150
Finland

Email: sahsan@cc.hut.fi

Lars Eggert
NetApp
Sonnenallee 1
Kirchheim 85551
Germany

Phone: +49 151 12055791
Email: lars@netapp.com
URI: <http://eggert.org/>

8. References

8.1. Normative References

- [1] Singh, V., Karkkainen, T., Ott, J., Ahsan, S., and L. Eggert, "Multipath RTP (MP RTP)", [draft-singh-avtcore-mprtp-04](#) (work in progress), February 2012.
- [2] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, [RFC 3550](#), July 2003.
- [3] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [4] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, [RFC 5234](#), January 2008.
- [5] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", [RFC 5245](#), April 2010.
- [6] Perkins, C. and M. Westerlund, "Multiplexing RTP Data and Control Packets on a Single Port", [RFC 5761](#), April 2010.

8.2. Informative References

- [7] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", [RFC 3264](#), June 2002.
- [8] 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.
- [9] Schulzrinne, H., Rao, A., Lanphier, R., Westerlund, M., and M. Stiemerling, "Real Time Streaming Protocol 2.0 (RTSP)", [draft-ietf-mmusic-rfc2326bis-29](#) (work in progress), March 2012.
- [10] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", [RFC 4566](#), July 2006.
- [11] Goldberg, J., Westerlund, M., and T. Zeng, "A Network Address Translator (NAT) Traversal mechanism for media controlled by Real-Time Streaming Protocol (RTSP)", [draft-ietf-mmusic-rtsp-nat-12](#) (work in progress), May 2012.
- [12] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", [BCP 72](#), [RFC 3552](#), July 2003.

Appendix A. Change Log

Note to the RFC-Editor: please remove this section prior to publication as an RFC.

A.1. Changes in [draft-singh-mmusic-mprtp-sdp-extension-00](#)

- o The document is created by splitting the [draft-singh-avtcore-mprtp-04](#) into 2 parts. The RTP related stuff is kept in the former while the SDP related discussion is moved to this new document.

Authors' Addresses

Varun Singh
Aalto University
School of Science and Technology
Otakaari 5 A
Espoo, FIN 02150
Finland

Email: varun@comnet.tkk.fi
URI: <http://www.netlab.tkk.fi/~varun/>

Joerg Ott
Aalto University
School of Science and Technology
Otakaari 5 A
Espoo, FIN 02150
Finland

Email: jo@comnet.tkk.fi

Teemu Karkkainen
Aalto University
School of Science and Technology
Otakaari 5 A
Espoo, FIN 02150
Finland

Email: teemuk@comnet.tkk.fi

Ralf Globisch
Fraunhofer HHI
Einsteinufer 37
Berlin D-10587
Germany

Email: ralf.globisch@gmail.com

Thomas Schierl
Fraunhofer HHI
Einsteinufer 37
Berlin D-10587
Germany

Phone: +49-30-31002-227
Email: ts@thomas-schierl.de

