

AVT Core Working Group
Internet-Draft
Intended status: Experimental
Expires: July 15, 2013

V. Singh
T. Karkkainen
J. Ott
S. Ahsan
Aalto University
L. Eggert
NetApp
January 11, 2013

Multipath RTP (MP RTP)
draft-singh-avtcore-mprtp-06

Abstract

The Real-time Transport Protocol (RTP) is used to deliver real-time content and, along with the RTP Control Protocol (RTCP), forms the control channel between the sender and receiver. However, RTP and RTCP assume a single delivery path between the sender and receiver and make decisions based on the measured characteristics of this single path. Increasingly, endpoints are becoming multi-homed, which means that they are connected via multiple Internet paths. Network utilization can be improved when endpoints use multiple parallel paths for communication. The resulting increase in reliability and throughput can also enhance the user experience. This document extends the Real-time Transport Protocol (RTP) so that a single session can take advantage of the availability of multiple paths between two endpoints.

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 July 15, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](http://trustee.ietf.org/license-info) 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
1.3.	Use-cases	5
2.	Goals	5
2.1.	Functional goals	5
2.2.	Compatibility goals	6
3.	RTP Topologies	6
4.	MPRTP Architecture	6
5.	Example Media Flow Diagrams	8
5.1.	Streaming use-case	8
5.2.	Conversational use-case	9
6.	MPRTP Functional Blocks	10
7.	Available Mechanisms within the Functional Blocks	11
7.1.	Session Setup	11
7.1.1.	Connectivity Checks	11
7.1.2.	Adding New or Updating Interfaces	11
7.1.3.	In-band vs. Out-of-band Signaling	11
7.2.	Expanding RTP	13
7.3.	Expanding RTCP	13
7.4.	Failure Handling and Teardown	13
8.	MPRTP Protocol	14
8.1.	Overview	14
8.1.1.	Gather or Discovering Candidates	15
8.1.2.	NAT Traversal	15
8.1.3.	Choosing between In-band (in RTCP) and Out-of-band (in SDP) Interface Advertisement	16
8.1.4.	In-band Interface Advertisement	16
8.1.5.	Subflow ID Assignment	16
8.1.6.	Active and Passive Subflows	16
8.2.	RTP Transmission	17
8.3.	Playout Considerations at the Receiver	17

8.4.	Subflow-specific RTCP Statistics and RTCP Aggregation . . .	17
8.5.	RTCP Transmission	18
9.	Packet Formats	18
9.1.	RTP Header Extension for MPRTTP	18
9.1.1.	MPRTTP RTP Extension for a Subflow	20
9.2.	RTCP Extension for MPRTTP (MPRTCP)	20
9.2.1.	MPRTCP Extension for Subflow Reporting	22
9.2.1.1.	MPRTCP for Subflow-specific SR, RR and XR	23
9.3.	MPRTCP Extension for Interface advertisement	25
10.	RTCP Timing reconsiderations for MPRTCP	26
11.	SDP Considerations	26
11.1.	Signaling MPRTTP Header Extension in SDP	27
11.2.	Signaling MPRTTP capability in SDP	27
11.3.	MPRTTP with ICE	28
11.4.	Increased Throughput	28
11.5.	Offer/Answer	28
11.5.1.	In-band Signaling Example	29
12.	IANA Considerations	30
12.1.	MPRTTP Header Extension	30
12.2.	MPRTCP Packet Type	30
12.3.	SDP Attributes	31
12.3.1.	"mprtp" attribute	31
13.	Security Considerations	32
14.	Acknowledgements	32
15.	References	32
15.1.	Normative References	32
15.2.	Informative References	33
Appendix A.	Interoperating with Legacy Applications	34
Appendix B.	Change Log	34
B.1.	Changes in draft-singh-avtcore-mprtp-05	34
B.2.	Changes in draft-singh-avtcore-mprtp-04	34
B.3.	Changes in draft-singh-avtcore-mprtp-03	35
B.4.	Changes in draft-singh-avtcore-mprtp-02	35
B.5.	Changes in draft-singh-avtcore-mprtp-01	35
Authors' Addresses	35

1. Introduction

Multi-homed endpoints are becoming common in today's Internet, e.g., devices that support multiple wireless access technologies such as 3G and Wireless LAN. This means that there is often more than one network path available between two endpoints. Transport protocols, such as RTP, have not been designed to take advantage of the availability of multiple concurrent paths and therefore cannot benefit from the increased capacity and reliability that can be achieved by pooling their respective capacities.

Multipath RTP (MPRTP) is an OPTIONAL extension to RTP [1] that allows splitting a single RTP stream into multiple subflows that are transmitted over different paths. In effect, this pools the resource capacity of multiple paths. Multipath RTCP (MPRTCP) is an extension to RTCP, it is used along with MPRTP to report per-path sender and receiver characteristics.

Other IETF transport protocols that are capable of using multiple paths include SCTP [12], MPTCP [13] and SHIM6 [14]. However, these protocols are not suitable for real-time communications.

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 [2].

1.2. Terminology

- o Endpoint: host either initiating or terminating an RTP flow.
- o Interface: logical or physical component that is capable of acquiring a unique IP address.
- o Path: sequence of links between a sender and a receiver. Typically, defined by a set of source and destination addresses.
- o Subflow: flow of RTP packets along a specific path, i.e., a subset of the packets belonging to an RTP stream. The combination of all RTP subflows forms the complete RTP stream. Typically, a subflow would map to a unique path, i.e., each combination of IP addresses and port pairs (5-tuple, including protocol) is a unique subflow.

1.3. Use-cases

The primary use-case for MPRTTP is transporting high bit-rate streaming multimedia content between endpoints, where at least one is multi-homed. Such endpoints could be residential IPTV devices that connect to the Internet through two different Internet service providers (ISPs), or mobile devices that connect to the Internet through 3G and WLAN interfaces. By allowing RTP to use multiple paths for transmission, the following gains can be achieved:

- o Higher quality: Pooling the resource capacity of multiple Internet paths allows higher bit-rate and higher quality codecs to be used. From the application perspective, the available bandwidth between the two endpoints increases.
- o Load balancing: Transmitting an RTP stream over multiple paths reduces the bandwidth usage on a single path, which in turn reduces the impact of the media stream on other traffic on that path.
- o Fault tolerance: When multiple paths are used in conjunction with redundancy mechanisms (FEC, re-transmissions, etc.), outages on one path have less impact on the overall perceived quality of the stream.

A secondary use-case for MPRTTP is transporting Voice over IP (VoIP) calls to a device with multiple interfaces. Again, such an endpoint could be a mobile device with multiple wireless interfaces. In this case, little is to be gained from resource pooling, i.e., higher capacity or load balancing, because a single path should be easily capable of handling the required load. However, using multiple concurrent subflows can improve fault tolerance, because traffic can shift between the subflows when path outages occur. This results in very fast transport-layer handovers that do not require support from signaling.

2. Goals

This section outlines the basic goals that multipath RTP aims to meet. These are broadly classified as Functional goals and Compatibility goals.

2.1. Functional goals

Allow unicast RTP session to be split into multiple subflows in order to be carried over multiple paths. This may prove beneficial in case of video streaming.

- o Increased Throughput: Cumulative capacity of the two paths may meet the requirements of the multimedia session. Therefore, MP RTP MUST support concurrent use of the multiple paths.
- o Improved Reliability: MP RTP SHOULD be able to send redundant packets or re-transmit packets along any available path to increase reliability.

The protocol SHOULD be able to open new subflows for an existing session when new paths appear and MUST be able to close subflows when paths disappear.

2.2. Compatibility goals

MP RTP MUST be backwards compatible; an MP RTP stream needs to fall back to be compatible with legacy RTP stacks if MP RTP support is not successfully negotiated.

- o Application Compatibility: MP RTP service model MUST be backwards compatible with existing RTP applications, i.e., an MP RTP stack MUST be able to work with legacy RTP applications and not require changes to them. Therefore, the basic RTP APIs MUST remain unchanged, but an MP RTP stack MAY provide extended APIs so that the application can configure any additional features provided by the MP RTP stack.
- o Network Compatibility: individual RTP subflows MUST themselves be well-formed RTP flows, so that they are able to traverse NATs and firewalls. This MUST be the case even when interfaces appear after session initiation. Interactive Connectivity Establishment (ICE) [3] MAY be used for discovering new interfaces or performing connectivity checks.

3. RTP Topologies

[RFC 5117](#) [15] describes a number of scenarios using mixers and translators in single-party (point-to-point), and multi-party (point-to-multipoint) scenarios. [RFC 3550](#) [1] ([Section 2.3](#) and 7.x) discuss in detail the impact of mixers and translators on RTP and RTCP packets. MP RTP assumes that if a mixer or translator exists in the network, then either all of the multiple paths or none of the multiple paths go via this component.

4. MP RTP Architecture

In a typical scenario, an RTP session uses a single path. In an

MPRTP scenario, an RTP session uses multiple subflows that each use a different path. Figure 1 shows the difference.

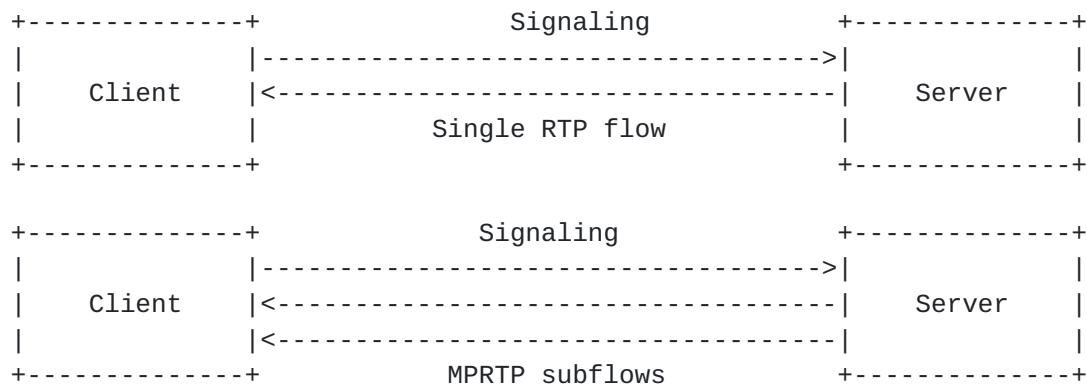


Figure 1: Comparison between traditional RTP streaming and MPRTTP

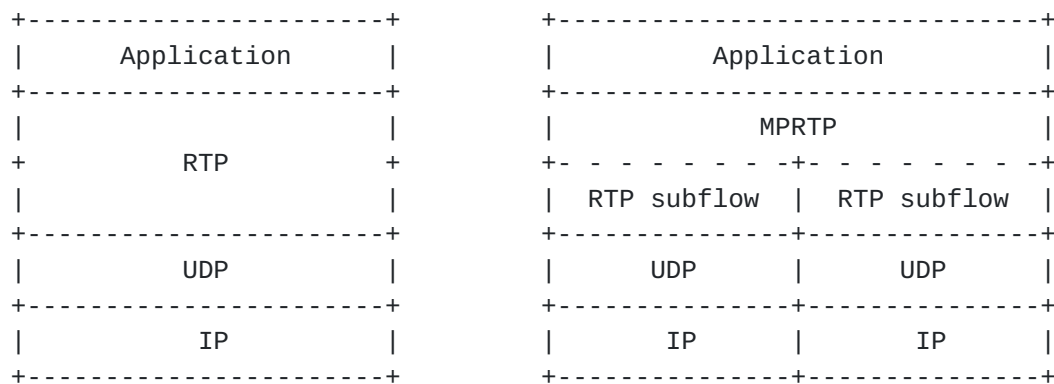


Figure 2: MP RTP Architecture

Figure 2 illustrates the differences between the standard RTP stack and the MP RTP stack. MP RTP receives a normal RTP session from the application and splits it into multiple RTP subflows. Each subflow is then sent along a different path to the receiver. To the network, each subflow appears as an independent, well-formed RTP flow. At the receiver, the subflows are combined to recreate the original RTP session. The MP RTP layer performs the following functions:

- 0 Path Management: The layer is aware of alternate paths to the other host, which may, for example, be the peer's multiple interfaces. This enables the endpoint to transmit differently marked packets along separate paths. MPRTTP also selects interfaces to send and receive data. Furthermore, it manages the port and IP address pair bindings for each subflow.

- o Packet Scheduling: the layer splits a single RTP flow into multiple subflows and sends them across multiple interfaces (paths). The splitting MAY BE done using different path characteristics.
- o Subflow recombination: the layer creates the original stream by recombining the independent subflows. Therefore, the multipath subflows appear as a single RTP stream to applications.

5. Example Media Flow Diagrams

There may be many complex technical scenarios for MP RTP, however, this memo only considers the following two scenarios: 1) a unidirectional media flow that represents the streaming use-case, and 2) a bidirectional media flow that represents a conversational use-case.

5.1. Streaming use-case

In the unidirectional scenario, the receiver (client) initiates a multimedia session with the sender (server). The receiver or the sender may have multiple interfaces and both endpoints are MP RTP-capable. Figure 3 shows this scenario. In this case, host A has multiple interfaces. Host B performs connectivity checks on host A's multiple interfaces. If the interfaces are reachable, then host B streams multimedia data along multiple paths to host A. Moreover, host B also sends RTCP Sender Reports (SR) for each subflow and host A responds with a normal RTCP Receiver Report (RR) for the overall session as well as the receiver statistics for each subflow. Host B distributes the packets across the subflows based on the individually measured path characteristics.

Alternatively, to reduce media startup time, host B may start streaming multimedia data to host A's initiating interface and then perform connectivity checks for the other interfaces. This method of updating a single path session to a multipath session is called "multipath session upgrade".

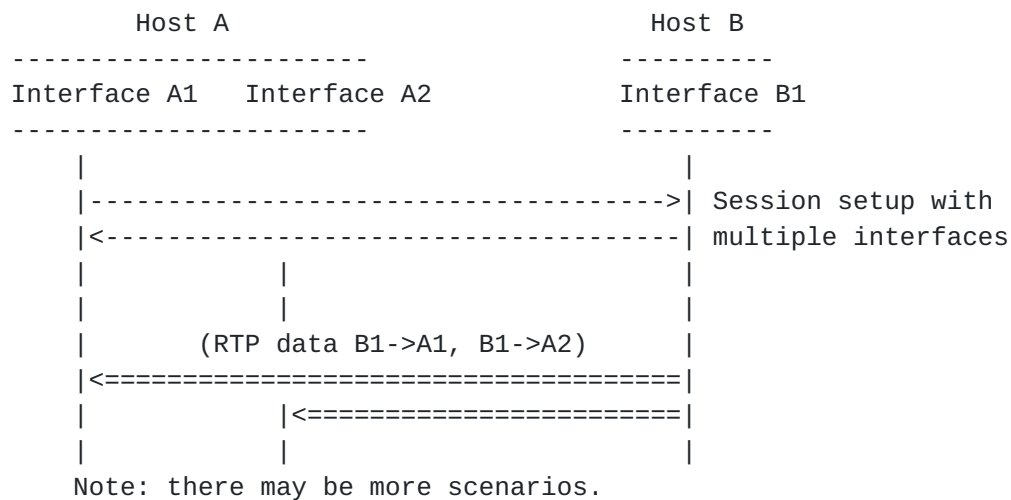


Figure 3: Unidirectional media flow

5.2. Conversational use-case

In the bidirectional scenario, multimedia data flows in both directions. The two hosts exchange their lists of interfaces with each other and perform connectivity checks. Communication begins after each host finds suitable address, port pairs. Interfaces that receive data send back RTCP receiver statistics for that path (based on the 5-tuple). The hosts balance their multimedia stream across multiple paths based on the per path reception statistics and its own volume of traffic. Figure 4 describes an example of a bidirectional flow.

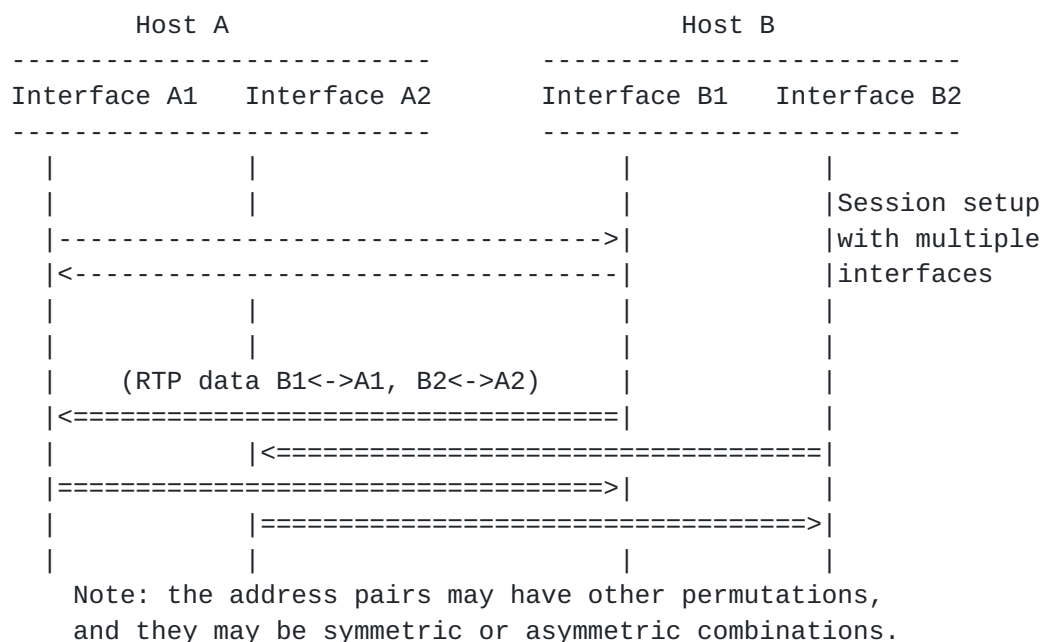


Figure 4: Bidirectional flow

6. MP RTP Functional Blocks

This section describes some of the functional blocks needed for MP RTP. We then investigate each block and consider available mechanisms in the next section.

1. **Session Setup:** Interfaces may appear or disappear at anytime during the session. To preserve backward compatibility with legacy applications, a multipath session **MUST** look like a bundle of individual RTP sessions. A multipath session may be upgraded from a typical single path session, as and when new interfaces appear or disappear. However, it is also possible to setup a multipath session from the beginning, if the interfaces are available at the start of the multimedia session.
2. **Expanding RTP:** For a multipath session, each subflow **MUST** look like an independent RTP flow, so that individual RTCP messages can be generated per subflow. Furthermore, MP RTP splits the single multimedia stream into multiple subflows based on path characteristics (e.g. RTT, loss-rate, receiver rate, bandwidth-delay product etc.) and dynamically adjusts the load on each link.
3. **Adding Interfaces:** Interfaces on the host need to be regularly discovered and advertised. This can be done at session setup and/or during the session. Discovering interfaces is outside the scope of this document.
4. **Expanding RTCP:** MP RTP **MUST** provide per path RTCP reports for monitoring the quality of the path, for load balancing, or for congestion control, etc. To maintain backward compatibility with legacy applications, the receiver **MUST** also send aggregate RTCP reports along with the per-path reports.
5. **Maintenance and Failure Handling:** In a multi-homed endpoint interfaces may appear and disappear. If this occurs at the sender, it has to re-adjust the load on the available links. On the other hand, if this occurs at the receiver, then the multimedia data transmitted by the sender to those interfaces is lost. This data may be re-transmitted along a different path i.e., to a different interface on the receiver. Furthermore, the endpoint has to either explicitly signal the disappearance of an interface, or the other endpoint has to detect it (by lack of media packets, RTCP feedback, or keep-alive packets).

6. Teardown: The MPRTTP layer releases the occupied ports on the interfaces.

7. Available Mechanisms within the Functional Blocks

This section discusses some of the possible alternatives for each functional block mentioned in the previous section.

7.1. Session Setup

MPRTTP 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) [16] in Session Initiation Protocol (SIP) [17], Real-Time Streaming Protocol (RTSP) [18]) or in-band signaling (e.g., RTP/RTCP header extension, Session Traversal Utilities for NAT (STUN) messages).

7.1.1. Connectivity Checks

The endpoint SHOULD be capable of performing connectivity checks (e.g., using ICE [3]). If the IP addresses of the endpoints are reachable (e.g., globally addressable, same network etc) then connectivity checks are not needed.

7.1.2. Adding New or Updating Interfaces

Interfaces can appear and disappear during a session, the endpoint MUST be capable of advertising the changes in its set of usable interfaces. Additionally, the application or OS may define a policy on when and/or what interfaces are usable. However, MPRTTP requires a method to advertise or notify the other endpoint about the updated set of usable interfaces.

7.1.3. In-band vs. Out-of-band Signaling

MTRTP nodes will generally use a signaling protocol to establish their MPRTTP session. With the existence of such a signaling relationship, two alternatives become available to exchange information about the available interfaces on each side for extending RTP sessions to MPRTTP and for modifying MPRTTP sessions: in-band and out-of-band signaling.

In-band signaling refers to using mechanisms of RTP/RTCP itself to communicate interface addresses, e.g., a dedicated RTCP extensions along the lines of the one defined to communicate information about the feedback target for RTP over SSM [4]. In-band signaling does not

rely on the availability of a separate signaling connection and the information flows along the same path as the media streams, thus minimizing dependencies. Moreover, if the media channel is secured (e.g., by means of SRTP), the signaling is implicitly protected as well if SRTCP encryption and authentication are chosen. In-band signaling is also expected to take a direct path to the peer, avoiding any signaling overlay-induced indirections and associated processing overheads in signaling elements -- avoiding such may be especially worthwhile if frequent updates may occur as in the case of mobile users. Finally, RTCP is usually sent sufficiently frequently (in point-to-point settings) to provide enough opportunities for transmission and (in case of loss) retransmission of the corresponding RTCP packets.

Examples for in-band signaling include RTCP extensions as noted above or suitable extensions to STUN [5].

Out-of-band signaling refers to using a separate signaling connection (via SIP, RTSP, or HTTP) to exchange interface information, e.g., expressed in SDP. Clear benefits are that signaling occurs at setup time anyway and that experience and SDP syntax (and procedures) are available that can be re-used or easily adapted to provide the necessary capabilities. In contrast to RTCP, SDP offers a reliable communication channel so that no separate retransmissions logic is necessary. In SDP, especially when combined with ICE, connectivity check mechanisms including sophisticated rules are readily available. While SDP is not inherently protected, suitable security may need to be applied anyway to the basic session setup.

Examples for out-of-band signaling are dedicated extensions to SDP; those may be combined with ICE.

Both types of mechanisms have their pros and cons for middleboxes. With in-band signaling, control packets take the same path as the media packets and they can be directly inspected by middleboxes so that the elements operating on the signaling channel do not need to understand new SDP. With out-of-band signaling, only the middleboxes processing the signaling need to be modified and those on the data forwarding path can remain untouched.

Overall, it may appear sensible to provide a combination of both mechanisms: out-of-band signaling for session setup and initial interface negotiation and in-band signaling to deal with frequent changes in interface state (and for the potential case, albeit rather theoretical case of MPRTCP communication without a signaling channel).

In its present version, this document explores both options to provide a broad understanding of how the corresponding mechanisms

would look like.

7.2. Expanding RTP

RTCP [1] is generated per media session. However, with MPRTTP, the media sender spreads the RTP load across several interfaces. The media sender SHOULD make the path selection, load balancing and fault tolerance decisions based on the characteristics of each path. Therefore, apart from normal RTP sequence numbers defined in [1], the MPRTTP sender MUST add subflow-specific sequence numbers to help calculate fractional losses, jitter, RTT, playout time, etc., for each path, and a subflow identifier to associate the characteristics with a path. The RTP header extension for MPRTTP is shown in [Section 9.1](#).

7.3. Expanding RTCP

To provide accurate per path information an MPRTTP endpoint MUST send (SR/RR) report for each unique subflow along with the overall session RTCP report. Therefore, the additional subflow reporting affects the RTCP bandwidth and the RTCP reporting interval. RTCP report scheduling for each subflow may cause a problem for RTCP recombination and reconstruction in cases when 1) RTCP for a subflow is lost, and 2) RTCP for a subflow arrives later than the other subflows. (There may be other cases as well.)

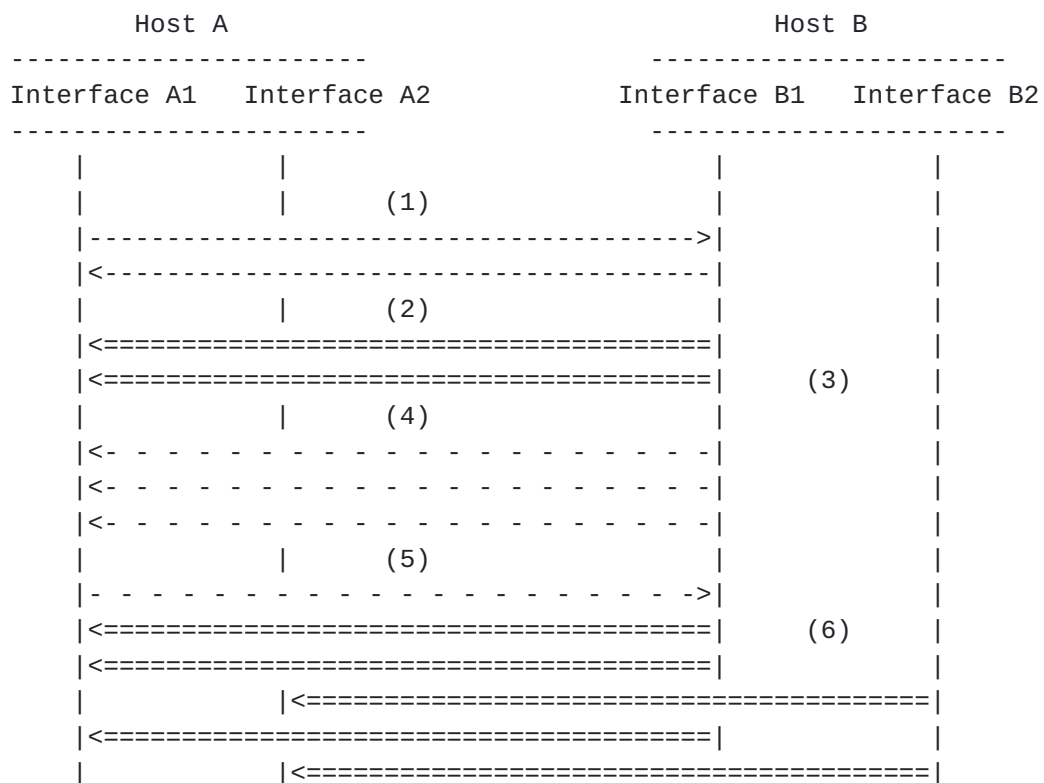
The sender distributes the media across different paths using the per path RTCP reports. However, this document doesn't cover algorithms for congestion control or load balancing.

7.4. Failure Handling and Teardown

An MPRTTP endpoint MUST keep alive subflows that have been negotiated but no media is sent on them. Moreover, using the information in the subflow reports, a sender can monitor an active subflow for failure (errors, unreachability, congestion) and decide not to use (make the active subflow passive), or teardown the subflow.

If an interface disappears, the endpoint MUST send an updated interface advertisement without the interface and release the the associated subflows.

8. MPRTP Protocol



Key:

| Interface

---> Signaling Protocol

<=== RTP Packets

- -> RTCP Packet

Figure 5: MPRTP New Interface

8.1. Overview

The bullet points explain the different steps shown in Figure 5 for upgrading a single path multimedia session to multipath session.

(1) The first two interactions between the hosts represents the establishment of a normal RTP session. This may performed e.g. using SIP or RTSP.

(2) When the RTP session has been established, host B streams media using its interface B1 to host A at interface A1.

(3) Host B supports sending media using MPRTP and becomes aware of an additional interface B2.

(4) Host B advertises the multiple interface addresses.

(5) Host A supports receiving media using MPRTTP and becomes aware of an additional interface A2.

Side note, even if an MPRTTP-capable host has only one interface, it MUST respond to the advertisement with its single interface.

(6) Each host receives information about the additional interfaces and the appropriate endpoints starts to stream the multimedia content using the additional paths.

If needed, each endpoint will need to independently perform connectivity checks (not shown in figure) and ascertain reachability before using the paths.

8.1.1. Gather or Discovering Candidates

The endpoint periodically polls the operating system or is notified when an additional interface appears. If the endpoint wants to use the additional interface for MPRTTP it MUST advertise it to the other peers. The endpoint may also use ICE [3] to gather additional candidates.

8.1.2. NAT Traversal

After gathering their interface candidates, the endpoints decide internally if they wish to perform connectivity checks.

[[Comment.1: Legacy applications do not require ICE for session establishment, therefore, MPRTTP should not require it as well.
--Editor]]

If the endpoint chooses to perform connectivity checks then it MUST first advertise the gathered candidates as ICE candidates in SDP during session setup and let ICE perform the connectivity checks. As soon as a sufficient number of connectivity checks succeed, the endpoint can use the successful candidates to advertise its MPRTTP interface candidates.

Alternatively, if the endpoint supports mobility extensions for ICE [5], it can let the ICE agent gather and perform the connectivity checks. When the connectivity checks succeed, the ICE agent should notify the MPRTTP layer of these new paths (5-tuples) so that these new paths can be used by MPRTTP to send media packets.

[Editors note: The interaction between the RTP agent and ICE agent is needed for implementing a scheduling algorithm or congestion

control.]

8.1.3. Choosing between In-band (in RTCP) and Out-of-band (in SDP) Interface Advertisement

If there is no media flowing at the moment and the application wants to use the interfaces from the start of the session, it should advertise them in SDP (See [6]). Alternatively, the endpoint can setup the session as a single path media session and upgrade the session to multipath by advertising the session in-band (See [Section 8.1.4](#) or [5]). Moreover, if an interface appears and disappears, the endpoint SHOULD prefer to advertise it in-band because the endpoint would not have to wait for a response from the other endpoint before starting to use the interface. However, if there is a conflict between an in-band and out-of-band advertisement, i.e., the endpoint receives an in-band advertisement while it has a pending out-of-band advertisement, or vice versa then the session is setup using out-of-band mechanisms.

8.1.4. In-band Interface Advertisement

To advertise the multiple interfaces in RTCP, an MPRTCP-capable endpoint MUST add the MPRTCP Interface Advertisement defined in Figure 13 with the RTCP Sender Report (SR). Each unique address is encapsulated in an Interface Advertisement block and contains the IP address, RTP and RTCP port addresses. The Interface Advertisement blocks are ordered based on a decreasing priority level. On receiving the MPRTCP Interface Advertisement, an MPRTCP-capable receiver MUST respond with the set of interfaces (subset or all available) it wants to use.

If the sender and receiver have only one interface, then the endpoints MUST indicate the negotiated single path IP, RTP port and RTCP port addresses.

8.1.5. Subflow ID Assignment

After interface advertisements have been exchanged, the endpoint MUST associate a Subflow ID to each unique subflow. Each combination of sender and receiver IP addresses and port pairs (5-tuple) is a unique subflow. If the connectivity checks have been performed then the endpoint MUST only use the subflows for which the connectivity checks have succeeded.

8.1.6. Active and Passive Subflows

To send and receive data an endpoint MAY use any number of subflows from the set of available subflows. The subflows that carry media

data are called active subflows, while those subflows that don't send any media packets (fallback paths) are called passive subflows.

An endpoint MUST multiplex the subflow specific RTP and RTCP packets on the same port to keep the NAT bindings alive. This is inline with the recommendation made in [RFC6263](#)[19]. Moreover, if an endpoint uses ICE, multiplexing RTP and RTCP reduces the number of components from 2 to 1 per media stream. If no MPRTCP or MPRTCP packets are received on a particular subflow at a receiver, the receiver SHOULD remove the subflow from active and passive lists and not send any MPRTCP reports for that subflow. It may keep the bindings in a dead-pool, so that if the 5-tuple or subflow reappears, it can quickly reallocate it based on past history.

8.2. RTP Transmission

If both endpoints are MPRTCP-capable and if they want to use their multiple interfaces for sending the media stream then they MUST use the MPRTCP header extensions. They MAY use normal RTP with legacy endpoints (see [Appendix A](#)).

An MPRTCP endpoint sends RTP packets with an MPRTCP extension that maps the media packet to a specific subflow (see Figure 8). The MPRTCP layer SHOULD associate an RTP packet with a subflow based on a scheduling strategy. The scheduling strategy may either choose to augment the paths to create higher throughput or use the alternate paths for enhancing resilience or error-repair. Due to the changes in path characteristics, the endpoint should be able change its scheduling strategy during an ongoing session. The MPRTCP sender MUST also populate the subflow specific fields described in the MPRTCP extension header (see [Section 9.1.1](#)).

8.3. Playout Considerations at the Receiver

A media receiver, irrespective of MPRTCP support or not, should be able to playback the media stream because the received RTP packets are compliant to [\[1\]](#), i.e., a non-MPRTCP receiver will ignore the MPRTCP header and still be able to playback the RTP packets. However, the variation of jitter and loss per path may affect proper playout. The receiver can compensate for the jitter by modifying the playout delay (i.e., by calculating skew across all paths) of the received RTP packets.

8.4. Subflow-specific RTCP Statistics and RTCP Aggregation

Aggregate RTCP provides the overall media statistics and follows the normal RTCP defined in [RFC3550](#) [\[1\]](#). However, subflow specific RTCP provides the per path media statistics because the aggregate RTCP

report may not provide sufficient per path information to an MP RTP scheduler. Specifically, the scheduler should be aware of each path's RTT and loss-rate, which an aggregate RTCP cannot provide. The sender/receiver MUST use non-compound RTCP reports defined in [RFC5506](#) [7] to transmit the aggregate and subflow-specific RTCP reports. Also, each subflow and the aggregate RTCP report MUST follow the timing rules defined in [8].

The RTCP reporting interval is locally implemented and the scheduling of the RTCP reports may depend on the the behavior of each path. For instance, the RTCP interval may be different for a passive path than an active path to keep port bindings alive. Additionally, an endpoint may decide to share the RTCP reporting bit rate equally across all its paths or schedule based on the receiver rate on each path.

[8.5.](#) RTCP Transmission

The sender sends an RTCP SR on each active path. For each SR the receiver gets, it echoes one back to the same IP address-port pair that sent the SR. The receiver tries to choose the symmetric path and if the routing is symmetric then the per-path RTT calculations will work out correctly. However, even if the paths are not symmetric, the sender would at maximum, under-estimate the RTT of the path by a factor of half of the actual path RTT.

[9.](#) Packet Formats

In this section we define the protocol structures described in the previous sections.

[9.1.](#) RTP Header Extension for MP RTP

The MP RTP header extension is used to distribute a single RTP stream over multiple subflows.

The header conforms to the one-byte RTP header extension defined in [9]. The header extension contains a 16-bit length field that counts the number of 32-bit words in the extension, excluding the four-octet extension header (therefore zero is a valid length, see Section 5.3.1 of [1] for details).

The RTP header for each subflow is defined below:

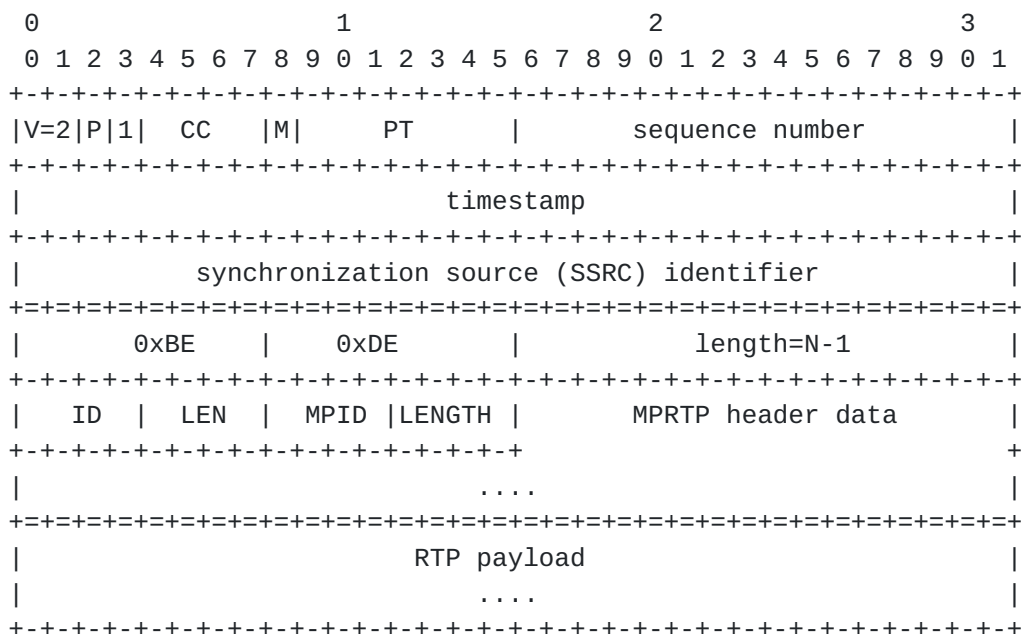


Figure 6: Generic MP RTP header extension

MPID:

The MPID field corresponds to the type of MP RTP packet.
Namely:

MPID ID	Use
Value	
0x0	Subflow RTP Header. For this case the Length is set to 4

Figure 7: RTP header extension values for MP RTP (H-Ext ID)

length

The 4-bit length field is the length of extension data in bytes not including the H-Ext ID and length fields. The value zero indicates there is no data following.

MP RTP header data

Carries the MPID specific data as described in the following sub-sections.

9.1.1. MPRTP RTP Extension for a Subflow

The RTP header for each subflow is defined below:

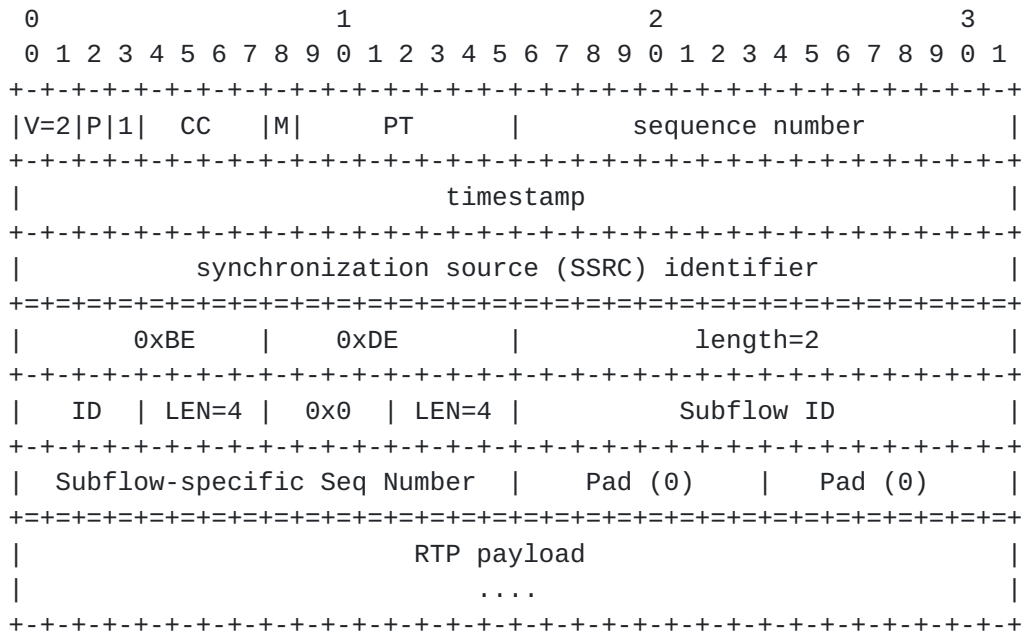


Figure 8: MPRTP header for subflow

MP ID = 0x0

Indicates that the MPRTP header extension carries subflow specific information.

length = 4

Subflow ID: Identifier of the subflow. Every RTP packet belonging to the same subflow carries the same unique subflow identifier.

Flow-Specific Sequence Number (FSSN): Sequence of the packet in the subflow. Each subflow has its own strictly monotonically increasing sequence number space.

9.2. RTCP Extension for MPRTP (MPRTCP)

The MPRTP RTCP header extension is used to 1) provide RTCP feedback per subflow to determine the characteristics of each path, and 2) advertise each interface.

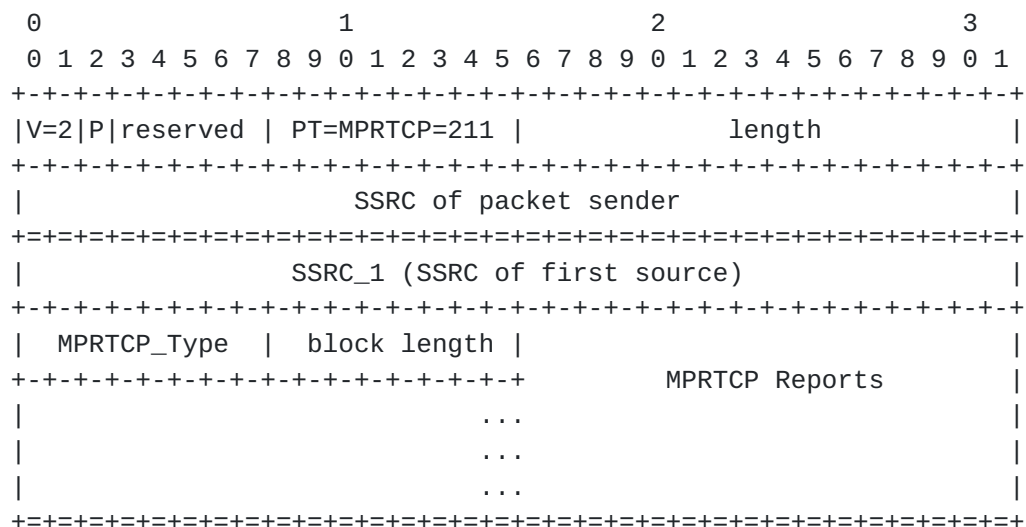


Figure 9: Generic RTCP Extension for MPRTCP (MPRTCP) [appended to normal SR/RR]

MPRTCP: 8 bits

Contains the constant 211 to identify this as an Multipath RTCP packet.

length: 16 bits

As described for the RTCP packet (see [Section 6.4.1](#) of the RTP specification [1]), the length of this is in 32-bit words minus one, including the header and any padding.

MPRTCP_Type: 8-bits

The MPRTCP_Type field corresponds to the type of MPRTCP RTCP packet. Namely:

MPRTCP_Type	Use
Value	
0	Subflow Specific Report
1	Interface Advertisement (IPv4 Address)
2	Interface Advertisement (IPv4 Address)
3	Interface Advertisement (DNS Address)

Figure 10: RTP header extension values for MPRTCP (MPRTCP_Type)

block length: 8-bits

The 8-bit length field is the length of the encapsulated MPRTCP reports in 32-bit word length not including the MPRTCP_Type and length fields. The value zero indicates there is no data following.

MPRTCP Reports: variable size

Defined later in 9.2.1 and 9.3.

9.2.1. MPRTCP Extension for Subflow Reporting

When sending a report for a specific subflow the sender or receiver MUST add only the reports associated with that 5-tuple. Each subflow is reported independently using the following MPRTCP Feedback header.

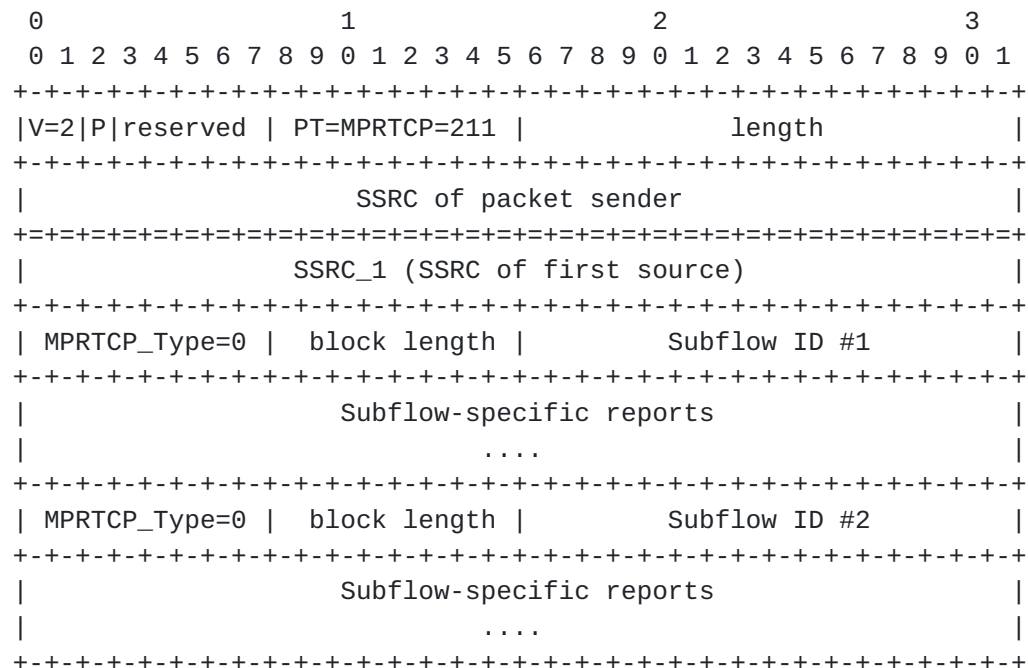


Figure 11: MPRTCP Subflow Reporting Header

MPRTCP_Type: 0

The value indicates that the encapsulated block is a subflow report.

block length: 8-bits

The 8-bit length field is the length of the encapsulated subflow-specific reports in 32-bit word length not including the

MPRTCP_Type and length fields.

Subflow ID: 16 bits

Subflow identifier is the value associated with the subflow the endpoint is reporting about. If it is a sender it MUST use the Subflow ID associated with the 5-tuple. If it is a receiver it MUST use the Subflow ID received in the Subflow-specific Sender Report.

Subflow-specific reports: variable

Subflow-specific report contains all the reports associated with the Subflow ID. For a sender, it MUST include the Subflow-specific Sender Report (SSR). For a receiver, it MUST include Subflow-specific Receiver Report (SRR). Additionally, if the receiver supports subflow-specific extension reports then it MUST append them to the SRR.

9.2.1.1. MPRTCP for Subflow-specific SR, RR and XR

[[Comment.2: inside the context of subflow specific reports can we reuse the payload type code for Sender Report (PT=200), Receiver Report (PT=201), Extension Report (PT=207). Transport and Payload specific RTCP messages are session specific and SHOULD be used as before. --Editor]]

Example:


```

      0              1              2              3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|V=2|P|reserved | PT=MPRTCP=211 |                length |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                SSRC of packet sender |
+===+===+===+===+===+===+===+===+===+===+===+===+===+===+===+===+
|                SSRC_1 (SSRC of first source) |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| MPRTCP_Type=0 | block length | Subflow ID #1 |
+===+===+===+===+===+===+===+===+===+===+===+===+===+===+===+===+
|V=2|P|    RC    | PT=SR=200 |                length |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                SSRC of sender |
+===+===+===+===+===+===+===+===+===+===+===+===+===+===+===+===+
|                NTP timestamp, most significant word |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                NTP timestamp, least significant word |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                RTP timestamp |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                subflow's packet count |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                subflow's octet count |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| MPRTCP_Type=0 | block length | Subflow ID #2 |
+===+===+===+===+===+===+===+===+===+===+===+===+===+===+===+===+
|V=2|P|    RC    | PT=RR=201 |                length |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                SSRC of packet sender |
+===+===+===+===+===+===+===+===+===+===+===+===+===+===+===+===+
| fraction lost | cumulative number of packets lost |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                extended highest sequence number received |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                inter-arrival jitter |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                last SR (LSR) |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                delay since last SR (DLSR) |
+===+===+===+===+===+===+===+===+===+===+===+===+===+===+===+===+
|                Subflow specific extension reports |
|                .... |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 12: Example of reusing RTCP SR and RR inside an MPRTCP header (Bi-directional use-case, in case of uni-directional flow the subflow will only send an SR or RR).

9.3. MPRTCP Extension for Interface advertisement

This sub-section defines the RTCP header extension for in-band interface advertisement by the receiver. The interface advertisement block describes a method to represent IPv4, IPv6 and generic DNS-type addresses in a block format. It is based on the sub-reporting block in [4]. The interface advertisement SHOULD immediately follow the Receiver Report. If the Receiver Report is not present, then it MUST be appended to the Sender Report.

The endpoint MUST advertise the interfaces it wants to use whenever an interface appears or disappears and also when it receives an Interface Advertisement.

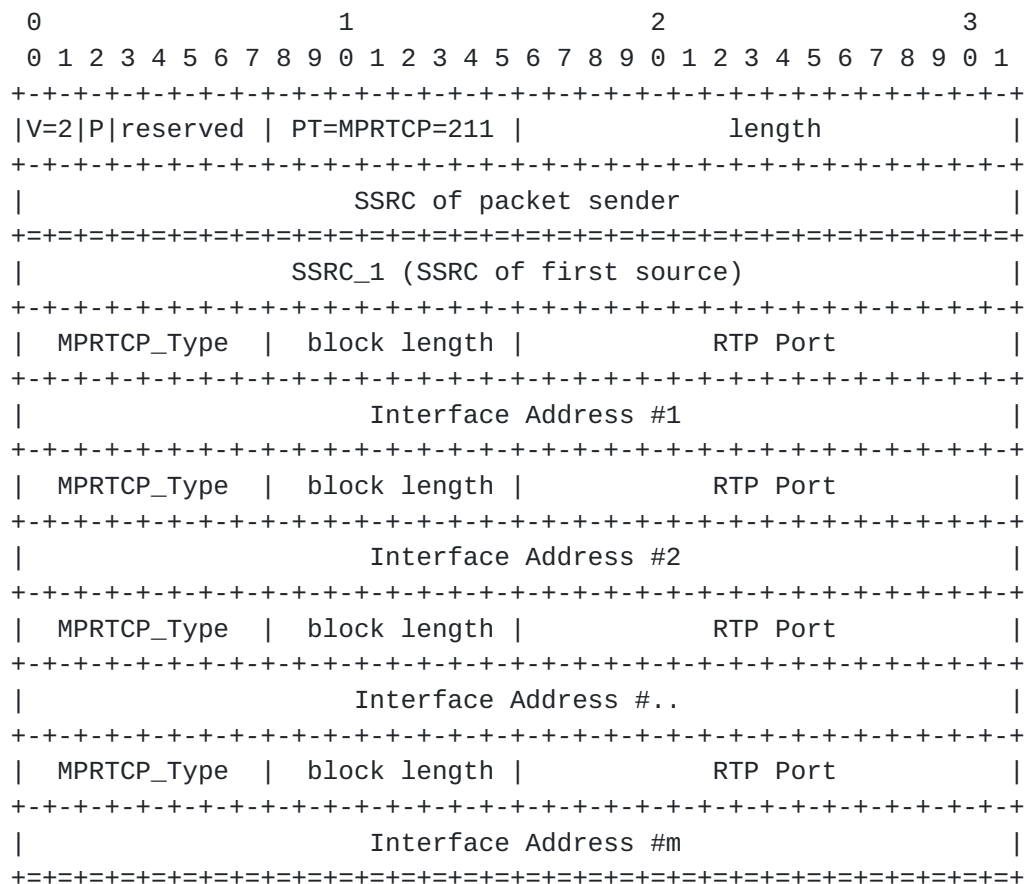


Figure 13: MPRTCP Interface Advertisement. (appended to SR/RR)

MPRTCP_Type: 8 bits

The MPRTCP_Type corresponds to the type of interface address.
Namely:

1: IPv4 address

2: IPv6 address

3: DNS name

block length: 8 bits

The length of the Interface Advertisement block in bytes.

For an IPv4 address, this should be 9 (i.e., 5 octets for the header and 4 octets for IPv4 address).

For an IPv6 address, this should be 21.

For a DNS name, the length field indicates the number of octets making up the string plus the 5 byte header.

RTP Port: 2 octets

The port number to which the sender sends RTP data. A port number of 0 is invalid and MUST NOT be used.

Interface Address: 4 octets (IPv4), 16 octets (IPv6), or n octets (DNS name)

The address to which receivers send feedback reports. For IPv4 and IPv6, fixed-length address fields are used. A DNS name is an arbitrary-length string. The string MAY contain Internationalizing Domain Names in Applications (IDNA) domain names and MUST be UTF-8 [[10](#)] encoded.

[10.](#) RTCP Timing reconsiderations for MPRTCP

MPRTCP endpoints MUST conform to the timing rule imposed in [[8](#)], i.e., the total RTCP rate between the participants MUST NOT exceed 5% of the media rate. For each endpoint, a subflow MUST send the aggregate and subflow-specific report. The endpoint SHOULD schedule the RTCP reports for the active subflows based on the share of the transmitted or received bit rate to the average media bit rate, this method ensures fair sharing of the RTCP bandwidth. Alternatively, the endpoint MAY schedule the reports in round-robin.

[11.](#) SDP Considerations

11.1. Signaling MPRTTP Header Extension in SDP

To indicate the use of the MPRTTP header extensions (see [Section 9](#)) in SDP, the sender MUST use the following URI in extmap:
urn:ietf:params:rtp-hdext:mprtp. This is a media level parameter.
Legacy RTP (non-MPRTTP) clients will ignore this header extension, but can continue to parse and decode the packet (see [Appendix A](#)).

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 H264/900000
a=fmtp:98 profile-level-id=42A01E;
a=extmap:1 urn:ietf:params:rtp-hdext:mprtp
```

11.2. Signaling MPRTTP capability in SDP

A participant of a media session MUST use SDP to indicate that it supports MPRTTP. Not providing this information will make the other endpoint ignore the RTCP extensions.

```
mprtp-attrb = "a=" "mprtp" [
    SP mprtp-optional-parameter]
    CRLF    ; flag to enable MPRTTP
```

The endpoint MUST use 'a=mprtp', if it is able to send and receive MPRTTP packets. Generally, senders and receivers MUST indicate this capability if they support MPRTTP and would like to use it in the specific media session being signaled. To exchange the additional interfaces, the endpoint SHOULD use the in-band signaling (See [Section 9.3](#)). Alternatively, advertise in SDP (See [\[6\]](#)).

MPRTTP endpoint multiplexes RTP and RTCP on a single port, sender MUST indicate support by adding "a=rtcp-mux" in SDP [\[11\]](#). If an endpoint receives an SDP without "a=rtcp-mux" but contains "a=mprtp", then the endpoint MUST infer support for multiplexing.

[[Comment.3: If a=mprtp is indicated, does the endpoint need to indicate a=rtcp-mux? because MPRTTP mandates RTP RTCP multiplexing.
--Editor]]

11.3. MP RTP with ICE

If the endpoints intend to use ICE [3] for discovering interfaces and running connectivity checks then the endpoint MUST advertise its ICE candidates in the initial OFFER, as defined in ICE [3]. Thereafter the endpoints run connectivity checks.

When an endpoint uses ICE's regular nomination [3] procedure, it chooses the best ICE candidate as the default path. In the case of an MP RTP endpoint, if more than one ICE candidate succeeded the connectivity checks then an MP RTP endpoint MAY advertise (some of) these in-band in RTCP as MP RTP interfaces.

When an endpoint uses ICE's aggressive nomination [3] procedure, the selected candidate may change as more ICE checks complete. Instead of sending updated offers as additional ICE candidates appear (transience), the endpoint it MAY use in-band signaling to advertise its interfaces, as defined in [Section 9.3](#).

If the default interface disappears and the paths used for MP RTP are different from the one in the c= and m= lines then the an alternate interface for which the ICE checks were successful 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 all its interfaces including 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 ICE candidates. Similarly, the initiating endpoint starts connectivity checks between the its interfaces (incl. the new interface) and all the received ICE candidates in the answer. If the connectivity checks succeed, the initiating endpoint MAY use in-band signaling (See [Section 9.3](#)) to advertise its new set of interfaces.

11.4. 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") [20] bandwidth.

11.5. Offer/Answer

When SDP [20] is used to negotiate MP RTP sessions following the offer/answer model [16], the "a=mp rtp" attribute (see [Section 11.2](#)) indicates the desire to use multiple interfaces to send or receive

media data. The initial SDP offer MUST include this attribute at the media level. If the answerer wishes to also use MP RTP, it MUST include a media-level "a=mp rtp" attribute in the answer. If the answer does not contain an "a=mp rtp" attribute, the offerer MUST NOT stream media over multiple paths and the offerer MUST NOT advertise additional MP RTP interfaces in-band or out-of-band.

When SDP is used in a declarative manner, the presence of an "a=mp rtp" 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.

11.5.1. 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=mp rtp
```

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
```

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) [6].

12. 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.

12.1. MPRTTP Header Extension

This document defines a new extension URI to the RTP Compact Header Extensions sub-registry of the Real-Time Transport Protocol (RTP) Parameters registry, according to the following data:

Extension URI: urn:ietf:params:rtp-hdext:mprrtp
Description: Multipath RTP
Reference: RFC XXXX

12.2. MPRTCP Packet Type

A new RTCP packet format has been registered with the RTCP Control Packet type (PT) Registry:

Name: MPRTCP
Long name: Multipath RTCP
Value: 211
Reference: RFC XXXX.

This document defines a substructure for MPRTCP packets. A new sub-registry has been set up for the sub-report block type (MPRTCP_Type) values for the MPRTCP packet, with the following registrations created initially:

Name:	Subflow Specific Report
Long name:	Multipath RTP Subflow Specific Report
Value:	0
Reference:	RFC XXXX.
Name:	IPv4 Address
Long name:	IPv4 Interface Address
Value:	1
Reference:	RFC XXXX.
Name:	IPv6 Address
Long name:	IPv6 Interface Address
Value:	2
Reference:	RFC XXXX.
Name:	DNS Name
Long name:	DNS Name indicating Interface Address
Value:	3
Reference:	RFC XXXX.

Further values may be registered on a first come, first served basis. For each new registration, it is mandatory that a permanent, stable, and publicly accessible document exists that specifies the semantics of the registered parameter as well as the syntax and semantics of the associated sub-report block. The general registration procedures of [20] apply.

12.3. SDP Attributes

This document defines a new SDP attribute, "mprtp", within the existing IANA registry of SDP Parameters.

12.3.1. "mprtp" attribute

- o Attribute Name: MP RTP
- 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 used to indicate support for Multipath RTP. It can also provide one of many possible interfaces for communication. These interface addresses may have been validated using ICE procedures.

- o Appropriate Values: See [Section 11.2](#) of RFC XXXX.

[13.](#) Security Considerations

TBD

All drafts are required to have a security considerations section. See [RFC 3552](#) [[21](#)] for a guide.

[14.](#) 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.

Thanks to Miguel A. Garcia, Ralf Globisch, Christer Holmberg, and Roni Even for providing valuable feedback on earlier versions of this draft

[15.](#) References

[15.1.](#) Normative References

- [1] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, [RFC 3550](#), July 2003.
- [2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [3] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", [RFC 5245](#), April 2010.
- [4] Ott, J., Chesterfield, J., and E. Schooler, "RTP Control Protocol (RTCP) Extensions for Single-Source Multicast Sessions with Unicast Feedback", [RFC 5760](#), February 2010.
- [5] Wing, D., Patil, P., Reddy, T., and P. Martinsen, "Mobility with ICE (MICE)", [draft-wing-mmusic-ice-mobility-02](#) (work in progress), October 2012.

- [6] Singh, V., Ott, J., Karkkainen, T., Globisch, R., and T. Schierl, "Multipath RTP (MPRTP) attribute in Session Description Protocol", [draft-singh-mmusic-mprtp-sdp-extension-01](#) (work in progress), January 2013.
- [7] Johansson, I. and M. Westerlund, "Support for Reduced-Size Real-Time Transport Control Protocol (RTCP): Opportunities and Consequences", [RFC 5506](#), April 2009.
- [8] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", [RFC 4585](#), July 2006.
- [9] Singer, D. and H. Desineni, "A General Mechanism for RTP Header Extensions", [RFC 5285](#), July 2008.
- [10] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, [RFC 3629](#), November 2003.
- [11] Perkins, C. and M. Westerlund, "Multiplexing RTP Data and Control Packets on a Single Port", [RFC 5761](#), April 2010.

15.2. Informative References

- [12] Stewart, R., "Stream Control Transmission Protocol", [RFC 4960](#), September 2007.
- [13] Ford, A., Raiciu, C., Handley, M., Barre, S., and J. Iyengar, "Architectural Guidelines for Multipath TCP Development", [RFC 6182](#), March 2011.
- [14] Nordmark, E. and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6", [RFC 5533](#), June 2009.
- [15] Westerlund, M. and S. Wenger, "RTP Topologies", [RFC 5117](#), January 2008.
- [16] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", [RFC 3264](#), June 2002.
- [17] 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.
- [18] Schulzrinne, H., Rao, A., Lanphier, R., Westerlund, M., and M. Stiemerling, "Real Time Streaming Protocol 2.0 (RTSP)", [draft-ietf-mmusic-rfc2326bis-30](#) (work in progress), July 2012.

- [19] Marjou, X. and A. Sollaud, "Application Mechanism for Keeping Alive the NAT Mappings Associated with RTP / RTP Control Protocol (RTCP) Flows", [RFC 6263](#), June 2011.
- [20] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", [RFC 4566](#), July 2006.
- [21] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", [BCP 72](#), [RFC 3552](#), July 2003.

[Appendix A.](#) Interoperating with Legacy Applications

An MPRTTP sender can use its multiple interfaces to send media to a legacy RTP client. The legacy receiver will ignore the subflow RTP header and the receiver's de-jitter buffer will try to compensate for the mismatch in per-path delay. However, the receiver can only send the overall or aggregate RTCP report which may be insufficient for an MPRTTP sender to adequately schedule packets or detect if a path disappeared.

An MPRTTP receiver can only use one of its interface when communicating with a legacy sender.

[Appendix B.](#) Change Log

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

[B.1.](#) Changes in [draft-singh-avtcore-mprtp-05](#)

- o SDP extensions moved to [draft-singh-mmusic-mprtp-sdp-extension-00](#). Kept only the basic 'a=mptrtp' attribute in this document.
- o Cleaned up ICE procedures for advertising only using in-band signaling.

[B.2.](#) Changes in [draft-singh-avtcore-mprtp-04](#)

- o Fixed missing 0xBEDE header in MPRTTP header format.
- o Removed connectivity checks and keep-alives from in-band signaling.
- o MPRTTP and MPRTCP are multiplexed on a single port.

- o MPRTCP packet headers optimized.
- o Made ICE optional
- o Updated Sections: 7.1.2, 8.1.x, 11.2, 11.4, 11.6.
- o Added how to use MPRTCP in RTSP ([Section 12](#)).
- o Updated IANA Considerations section.

B.3. Changes in [draft-singh-avtcore-mprtp-03](#)

- o Added this change log.
- o Updated [section 6](#), 7 and 8 based on comments from MMUSIC.
- o Updated [section 11](#) (SDP) based on comments of MMUSIC.
- o Updated SDP examples with ICE and non-ICE in out-of-band signaling scenario.
- o Added [Appendix A](#) on interop with legacy.
- o Updated IANA Considerations section.

B.4. Changes in [draft-singh-avtcore-mprtp-02](#)

- o MPRTCP protocol extensions use only one PT=210, instead of 210 and 211.
- o RTP header uses 1-byte extension instead of 2-byte.
- o Added section on RTCP Interval Calculations.
- o Added "mprtp-interface" attribute in SDP considerations.

B.5. Changes in [draft-singh-avtcore-mprtp-01](#)

- o Added MPRTCP and MPRTCP protocol extensions and examples.
- o WG changed from -avt to -avtcore.

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/>

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

Email: teemuk@comnet.tkk.fi

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

Email: jo@comnet.tkk.fi

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/>