

AVT  
Internet-Draft  
Intended status: Standards Track  
Expires: February 13, 2011

A. Begen  
D. Wing  
Cisco  
T. VanCaenegem  
Alcatel-Lucent  
August 12, 2010

Token-Based Port Mapping Between Unicast and Multicast RTP Sessions  
draft-begen-avt-token-for-portmapping-01

Abstract

This document presents an alternative port mapping solution that allows RTP receivers to choose their own ports for an auxiliary unicast session in RTP applications using both unicast and multicast services (almost) without the need for retrieving pre-authorization.

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 February 13, 2011.

Copyright Notice

Copyright (c) 2010 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 . . . . .	3
2. Requirements Notation . . . . .	4
3. Token-Based Port Mapping . . . . .	5
3.1. Token Request and Retrieval . . . . .	5
3.2. Unicast Session Establishment . . . . .	5
4. The portmapping-req Attribute . . . . .	9
5. Message Formats . . . . .	10
6. Procedures for Token Construction . . . . .	11
7. Validating Tokens . . . . .	12
8. SDP Example . . . . .	13
9. Address Pooling NATs . . . . .	15
10. Security Considerations . . . . .	16
11. IANA Considerations . . . . .	17
11.1. Registration of SDP Attributes . . . . .	17
12. Acknowledgments . . . . .	18
13. References . . . . .	19
13.1. Normative References . . . . .	19
13.2. Informative References . . . . .	19
Authors' Addresses . . . . .	21

## 1. Introduction

[I-D.ietf-avt-ports-for-ucast-mcast-rtp] provides several scenarios for RTP applications that use one or more unicast and multicast RTP sessions together. These applications require a Port Mapping solution that allows receivers to choose their desired UDP ports for RTP and RTCP in unicast session(s). There is an inherent delay in learning the public port mapping and signaling it with the Offer/Answer Model [RFC3264]. Thus, the receiver might wish to convey its port number(s) through a different mechanism.

[I-D.ietf-avt-ports-for-ucast-mcast-rtp] offers a Cookie-based solution. This memo presents a more lightweight solution, which we call the Token solution.

Following the same convention with

[I-D.ietf-avt-ports-for-ucast-mcast-rtp], we will refer to the RTP endpoints that serve other RTP endpoints over a unicast session as the Servers, and the receiving RTP endpoints as Clients.

## 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. Token-Based Port Mapping

Token-based Port Mapping consists of two steps: Token request and retrieval, and unicast session establishment. These are described in the following sections.

#### 3.1. Token Request and Retrieval

The first step is required to be completed only once. Once a Token is retrieved from a particular server, it may be used for all the unicast sessions the client will be running with this particular server. By default, Tokens are server specific. However, the client can use the same Token to communicate with different servers if these servers are provided with the same key used to generate the Token. The Token may become invalid if client's public IP address changes or when the server expires the token. In those cases, the client has to request a new Token.

The Token is essentially an opaque encapsulation that conveys client's IP address information (as seen by the server) using a reversible transform only known to the server. When a request is received, the server creates a Token for this particular client, and sends it back to the client. Later, when the client wants to establish a unicast session, the Token will be validated by the server, making sure that the IP address information matches. This is effective against DoS attacks, i.e., an attacker cannot simply spoof another client's IP address and start possibly a high-bitrate unicast transmission [I-D.ietf-avt-rapid-acquisition-for-rtp] towards random clients.

#### 3.2. Unicast Session Establishment

We illustrate the second step on the same example presented in [I-D.ietf-avt-ports-for-ucast-mcast-rtp].

Consider an SSM distribution network where a distribution source multicasts RTP packets to a large number of clients, and one or more retransmission servers function as feedback targets to collect unicast RTCP feedback from these clients [RFC5760]. The retransmission servers also join the primary multicast session to receive the multicast packets and cache them for a certain time period. When a client detects missing packets in the primary multicast session, it requests a retransmission from one of the retransmission servers by using an RTCP NACK message [RFC4585]. The retransmission server pulls the requested packet(s) out of the cache and retransmits them to the requesting client [RFC4588].

The pertaining RTP and RTCP flows are sketched in Figure 1. Between

the client and server, there may be one or more NAT devices [RFC4787].

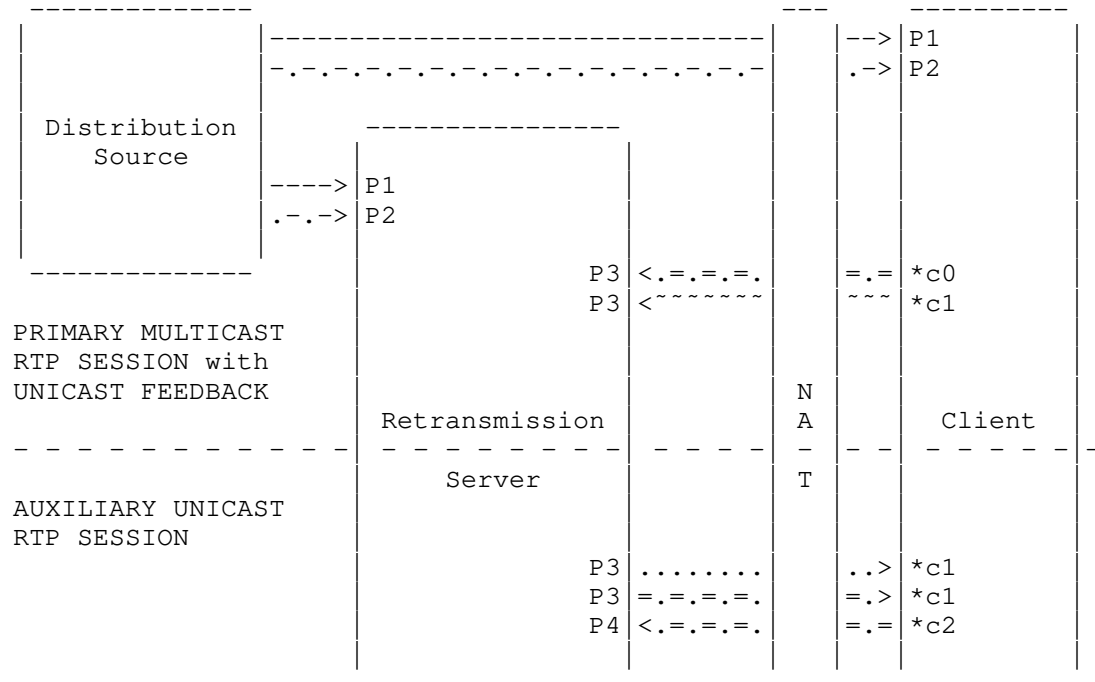


Figure 1: Example scenario showing an SSM distribution with support for retransmissions from a server

In this figure, we have the following multicast and unicast ports:

- o Ports P1 and P2 denote the destination RTP and RTCP ports in the primary multicast session, respectively. The clients listen to these ports to receive the multicast RTP and RTCP packets. Ports P1 and P2 are defined declaratively.
- o Port P3 denotes the RTCP port on the feedback target running on the retransmission server to collect the RTCP feedback messages,

and RTCP receiver and extended reports from the clients in the primary multicast session. This is also the port that the retransmission server uses to send the RTP packets and RTCP sender reports in the unicast session. Port P3 is defined declaratively.

- o Port P4 denotes the RTCP port on the retransmission server used to collect the RTCP receiver and extended reports for the unicast session. Port P4 is defined declaratively and MUST be different from port P3.
- o Ports \*c0, \*c1 and \*c2 are chosen by the client. \*c0 denotes the port on the client used to send the RTCP reports for the primary multicast session. \*c1 denotes the port on the client used to send the unicast RTCP feedback in the primary multicast session and to receive the RTP packets and RTCP sender reports in the unicast session. \*c2 denotes the port on the client used to send the RTCP receiver and extended reports in the unicast session. Ports c0, c1 and c2 MAY be the same port or different ports. However, there are two advantages of using the same port for both c0 and c1:
  1. Some NATs only keep bindings active when a packet goes from the inside to the outside of the NAT (See REQ-6 of Section 4.3 of [RFC4787]). Long RTP bursts may exceed that timeout. If c0=c1, the occasional RTCP receiver reports sent from port c0 will ensure the NAT does not time out the public port associated with the incoming RTP burst to port c1.
  2. Having c0=c1 conserves NAT port bindings.

Thus, it is strongly RECOMMENDED that c0=c1.

Once the server receives the RTCP NACK, the server sends the RTP burst to the IP address and UDP port the RTCP NACK came from.

In addition to the ports, we use the following notation:

- o DS: IP address of the distribution source
- o G: Destination multicast address
- o S: IP address of the retransmission server
- o C: IP address of the client
- o C': Public IP address of the client (as seen by the server)

We assume that the information declaratively defined is available as part of the session description information and is provided to the

clients. The Session Description Protocol (SDP) [RFC4566] and other session description methods can be used for this purpose.

The following steps summarize the Token-based solution:

1. The client ascertains server address (S) and port numbers (P3 and P4) from the session description.
2. The client determines its port numbers (\*c0, \*c1 and \*c2).
3. If the client does not have a valid Token for this particular server:
  - A. The client first sends a message to the server via a new RTCP message, called PortMappingRequest. This message can be sent from any port on the client side. The server learns client's public IP address (C') from the received message.

NOTE: The client can send this message anytime it wants (e.g., during initialization), and does not normally ever need to re-send this message (See Section 7).
  - B. The server generates an opaque encapsulation (called Token) that conveys client's IP address information using a reversible transform only known to the server. See Section 6.
  - C. The server sends the Token back to the client using a new RTCP message, called PortMappingResponse. This message **MUST** be sent from the port at which the server received the request.
4. The client includes the Token when necessary in the subsequent messages sent to the server. Note that the unicast session is only established after the server has received a feedback message (along with a valid Token) from the client for which it needs to react by sending unicast data. Until a unicast session is established, neither the server nor the client needs to send RTCP reports for the unicast session.
5. Normal flows ensue as shown in Figure 1. If the client uses the same port for both c0 and c1, the RTCP receiver and extended reports sent for the primary multicast session keep the P3->c1 binding alive. If the client uses different ports for c0 and c1, an explicit keep-alive message [I-D.ietf-avt-app-rtp-keepalive] may be needed to keep the P3->c1 binding alive during the lifetime of the unicast session, if that unicast session's lifetime exceeds the NAT's mapping refresh time.



#### 4. The portmapping-req Attribute

This new SDP attribute is used declaratively to indicate the port for obtaining a Token. Its presence also indicates that a Token **MUST** be included in the feedback messages sent to the server.

The formal description of the 'portmapping-req' attribute is defined by the following ABNF [RFC5234] syntax:

```
portmapping-req-attribute = "a=portmapping-req:" port CRLF
```

Here, the 'port' token is defined as specified in Section 9 of [RFC4566].

The 'portmapping-req' attribute **MAY** be used as a media-level attribute; it **MUST NOT** be used as a session-level attribute.

## 5. Message Formats

Editor's note: This section will define the message formats for requesting a Token (PortMappingRequest) and sending a Token (PortMappingResponse).

TBC.

## 6. Procedures for Token Construction

Editor's notes:

The Token may contain

- o Client's IP address
- o A timestamp to protect against replay attacks
- o HMAC [RFC2104] of the above information (where only the server knows the HMAC secret)

The server conveys the expiration date in the clear to the client via the PortMappingResponse message. Thus, the client can request a new Token before the current one expires.

Details are TBC.

## 7. Validating Tokens

Upon receipt of an RTCP feedback message containing a Token, the server validates the Token. The server considers a Token valid if the source IP address of the RTCP feedback message matches the IP address in the Token, and if the Token has not expired.

The IP address is encoded into the Token by the server, using an algorithm known only to the server. This, combined with the expiration, provides protection against DoS attacks so that a client using a certain IP address cannot cause one or more RTP packets to be sent to another client with a different IP address.

## 8. SDP Example

The SDP describing the scenario given in Figure 1 can be written as:

```
v=0
o=ali 1122334455 1122334466 IN IP4 nack.example.com
s=Local Retransmissions
t=0 0
a=group:FID 1 2
a=rtcp-unicast:rsi
m=video 41000 RTP/AVPF 98
i=Primary Multicast Stream
c=IN IP4 233.252.0.2/255
a=source-filter:incl IN IP4 233.252.0.2 198.51.100.1 ; Note 1
a=rtpmap:98 MP2T/90000s
a=multicast-rtcp:41500 ; Note 1
a=rtcp:42000 IN IP4 192.0.2.1 ; Note 2
a=rtcp-fb:98 nack ; Note 2
a=mid:1
m=video 42000 RTP/AVPF 99 ; Note 3
i=Unicast Retransmission Stream
c=IN IP4 192.0.2.1
a=rtpmap:99 rtx/90000
a=rtcp:42500 ; Note 4
a=fmtp:99 apt=98; rtx-time=5000
a=portmapping-req:30000 ; Note 5
a=mid:2
```

Figure 2: SDP describing an SSM distribution with support for retransmissions from a local server

In this SDP, we highlight the following notes:

Note 1: The source stream is multicast from a distribution source with a source IP address of 198.51.100.1 (DS) to the multicast destination address of 233.252.0.2 (G) and port 41000 (P1). The associated RTCP packets are multicast in the same group to port 41500 (P2).

Note 2: A retransmission server including feedback target functionality with an IP address of 192.0.2.1 (S) and port of 42000 (P3) is specified with the 'rtcp' attribute. The feedback functionality is enabled for the RTP stream with payload type 98 through the 'rtcp-fb' attribute [RFC4585].

Note 3: The port specified in the second "m" line (for the unicast stream) does not mean anything in this scenario as the client does not send any RTP traffic back to the server. To make this clear, we

might mandate to use the discard port in this line.

Note 4: The server uses port 42500 (P4) for the unicast sessions.

Note 5: The "a=portmapping-req" line indicates that a Token MUST be retrieved first before a unicast session can be established and that the Token request MUST be sent to port 30000.

## 9. Address Pooling NATs

Large-scale NAT (LSN) devices have a pool of public IPv4 addresses and map internal hosts to one of those public IPv4 addresses. As long as an internal host maintains an active mapping in the NAT, the same IPv4 address is assigned to new connections. However, once all of the host's mappings have been deleted (e.g., because of timeout), it is possible that a new connection from that same host will be assigned a different IPv4 address from the pool. When that occurs, the Token will be considered invalid by the server, causing an additional round trip for the client to acquire a fresh token.

Any traffic from the host which traverses the NAT will prevent this problem. As the host is sending RTCP receiver reports at least every 5 seconds (Section 6.2 of [RFC3550]) for the multicast session it is receiving, those RTCP messages will be sufficient to prevent this problem.

## 10. Security Considerations

The Token, which is generated based on a client's IP address and expiration date, provides protection against DoS attacks. An attacker using a certain IP address cannot cause one or more RTP packets to be sent to a victim client who has a different IP address.



## 11. IANA Considerations

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

Ali Begen  
abegen@cisco.com

### 11.1. Registration of SDP Attributes

This document registers a new attribute name in SDP.

```
SDP Attribute ("att-field"):  
Attribute name:    portmapping-req  
Long form:        Port for requesting Token  
Type of name:     att-field  
Type of attribute: Media level  
Subject to charset: No  
Purpose:          See this document  
Reference:        This document  
Values:           See this document
```

## 12. Acknowledgments

The approach presented in this document came out after discussions with various individuals in the AVT and MMUSIC WGs, and the breakout session held in the Anaheim meeting. We thank each of these individuals.

## 13. References

### 13.1. Normative 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.
- [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.
- [RFC4585] 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.
- [RFC5760] Ott, J., Chesterfield, J., and E. Schooler, "RTP Control Protocol (RTCP) Extensions for Single-Source Multicast Sessions with Unicast Feedback", RFC 5760, February 2010.
- [RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

### 13.2. Informative References

- [I-D.ietf-avt-ports-for-ucast-mcast-rtp]  
Begen, A. and B. Steeg, "Port Mapping Between Unicast and Multicast RTP Sessions",  
draft-ietf-avt-ports-for-ucast-mcast-rtp-02 (work in progress), May 2010.
- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.
- [I-D.ietf-avt-rapid-acquisition-for-rtp]  
Steeg, B., Begen, A., Caenegem, T., and Z. Vax, "Unicast-Based Rapid Acquisition of Multicast RTP Sessions",  
draft-ietf-avt-rapid-acquisition-for-rtp-11 (work in progress), July 2010.
- [RFC4787] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, January 2007.

- [RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R. Hakenberg, "RTP Retransmission Payload Format", RFC 4588, July 2006.
- [I-D.ietf-avt-app-rtp-keepalive]  
Marjou, X. and A. Sollaud, "Application Mechanism for keeping alive the Network Address Translator (NAT) mappings associated to RTP flows.", draft-ietf-avt-app-rtp-keepalive-08 (work in progress), June 2010.
- [RFC5506] Johansson, I. and M. Westerlund, "Support for Reduced-Size Real-Time Transport Control Protocol (RTCP): Opportunities and Consequences", RFC 5506, April 2009.
- [RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

Authors' Addresses

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

Email: [abegen@cisco.com](mailto:abegen@cisco.com)

Dan Wing  
Cisco Systems, Inc.  
170 West Tasman Dr.  
San Jose, CA 95134  
USA

Email: [dwing@cisco.com](mailto:dwing@cisco.com)

Tom VanCaenegem  
Alcatel-Lucent  
Copernicuslaan 50  
Antwerpen, 2018  
Belgium

Email: [Tom.Van\\_Caenegem@alcatel-lucent.be](mailto:Tom.Van_Caenegem@alcatel-lucent.be)



Audio/Video Transport Working Group  
Internet Draft  
Intended status: Informational  
Expires: April 2011

H.M. Stokking  
M.O. van Deventer  
O.A. Niamut  
F.A. Walraven  
R. van Brandenburg  
TNO Netherlands  
I. Vaishnavi  
CWI Netherlands  
October 11, 2010

RTCP XR Block Type for inter-destination media synchronization  
draft-brandenburg-avt-rtcp-for-idms-02.txt

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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

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

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

This Internet-Draft will expire on April 11, 2011.

Copyright Notice

Copyright (c) 2010 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.

## Abstract

This document specifies an RTCP XR Block Type and associated SDP parameter for inter-destination media synchronization (IDMS). The RTCP Block Type is used to collect media play-out information from participants in a group watching a specific RTP media stream and to distribute a summary of the collected information so that the participants can synchronize play-out.

Typical applications for inter-destination media synchronization are social TV, watching apart together and shared service control (i.e. applications where two or more geographically separated users are watching a media stream together).

## Table of Contents

1. Introduction.....	2
1.1. Inter-destination Media Synchronization.....	2
1.2. Applicability of RTCP to IDMS.....	3
1.3. Applicability of SDP to IDMS.....	3
1.4. This document and ETSI TISPAN.....	4
2. Inter-destination media synchronization use cases.....	4
2.1. Watching Apart Together.....	4
2.2. Shared Service Control.....	4
3. Architecture for inter-destination media synchronization....	4
3.1. Media Synchronization Application Server (MSAS).....	5
3.2. Synchronization Client (SC).....	5
3.3. Mixer.....	5
4. RTCP XR Block Type for IDMS.....	5
5. SDP parameter for IDMS.....	8
6. Security Considerations.....	9
7. IANA Considerations.....	9
8. Conclusions.....	9
9. References.....	10
9.1. Normative References.....	10
9.2. Informative References.....	10
10. Acknowledgments.....	10

## 1. Introduction

### 1.1. Inter-destination Media Synchronization

Inter-destination media synchronization (IDMS) refers to the play-out of media streams at two or more geographically distributed locations in a temporally synchronized manner. It can be applied to both unicast and multicast media streams and can be applied to any type and/or combination of streaming media, such as audio, video and text



(subtitles). [Boronat2009] provides an overview of technologies and algorithms for IDMS.

IDMS requires the exchange of information on media receipt and playout times. It may also require signaling for the initiation and maintenance of IDMS sessions and groups.

The presented RTCP-XR specification for IDMS is independent of the used synchronization algorithm, which is out-of-scope of this document.

### 1.2. Applicability of RTCP to IDMS

RTP and RTCP [RFC3550] are protocols that are typically used in conjunction. RTP (Real-time Transport Protocol) provides end-to-end network transport functions suitable for applications requiring real-time data transport, such as audio, video or data, over multicast or unicast network services.

The data transport is augmented by a control protocol (RTCP) to allow monitoring of the data delivery in a manner that is scalable to large multicast networks, and to provide minimal control and identification functionality.

RTP receivers and - senders provide reception quality feedback by sending out RTCP Receiver Report (RR) and Sender Report (SR) packets [RFC3550], which may be augmented by eXtended Reports (XR) [RFC3611].

IDMS involves the collection, summarizing and distribution of RTP packet arrival and play-out times.

RTCP is applicable to IDMS, as information on RTP packet arrival times and play-out times can be considered reception quality feedback information.

### 1.3. Applicability of SDP to IDMS

RTCP XR [RFC3611] defines the Extended Report (XR) packet type for the RTP Control Protocol (RTCP), and defines how the use of XR packets can be signaled by an application using the Session Description Protocol (SDP).

SDP signaling is used to set up and maintain a synchronization group between Synchronization Clients (SCs).

#### 1.4. This document and ETSI TISPAN

ETSI TISPAN [TS 183 063] has specified architecture and protocol for IDMS using RTCP XR exchange and optional SDP signaling. This internet draft is an excerpt of the IDMS part of that specification.

### 2. Inter-destination media synchronization use cases

Social TV is the combination of media content consumption by two or more users at different devices and locations and real-time communication between those users.

#### 2.1. Watching Apart Together

Watching Apart Together is an example of Social TV, where two or more users watch the same television broadcast at different devices and locations, while communicating with each other using text, audio and/or video.

A skew in the media play-out of the two or more users can have adverse effects on their experience. A well-known use case here is one friend experiencing a goal in a football match well before or after other friend(s). Thus IDMS is required to provide play-out synchronization.

#### 2.2. Shared Service Control

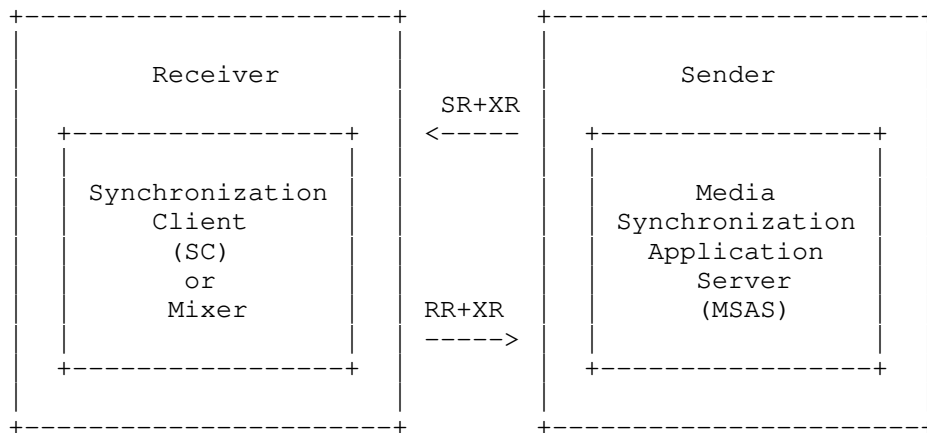
Shared Service Control is an example of Social TV, where two or more users experience some content-on-demand together, while sharing the trick-play controls (play, pause, fast forward, rewind) of the content on demand.

Similar to the previous use case, without IDMS, differences in play-out speed and the effect of transit delay of trick-play control signals would desynchronize content play-out.

### 3. Architecture for inter-destination media synchronization

The architecture for IDMS, which is based around a sync-maestro architecture [Boronat2009], is sketched below. The SC and MSAS entities are shown as additional functionality for the RTP receiver and sender respectively.

It should be noted that a master-slave type of architecture is also supported by having one of the SC devices also act as an MSAS.



### 3.1. Media Synchronization Application Server (MSAS)

An MSAS collects RTP packet arrival times and play-out times from one or more SC(s) in a synchronization group. The MSAS summarizes and distributes this information to the SCs in the synchronization group, e.g. by determining the SC with the most lagged play-out and using its reported RTP packet arrival time and play-out time as a summary.

### 3.2. Synchronization Client (SC)

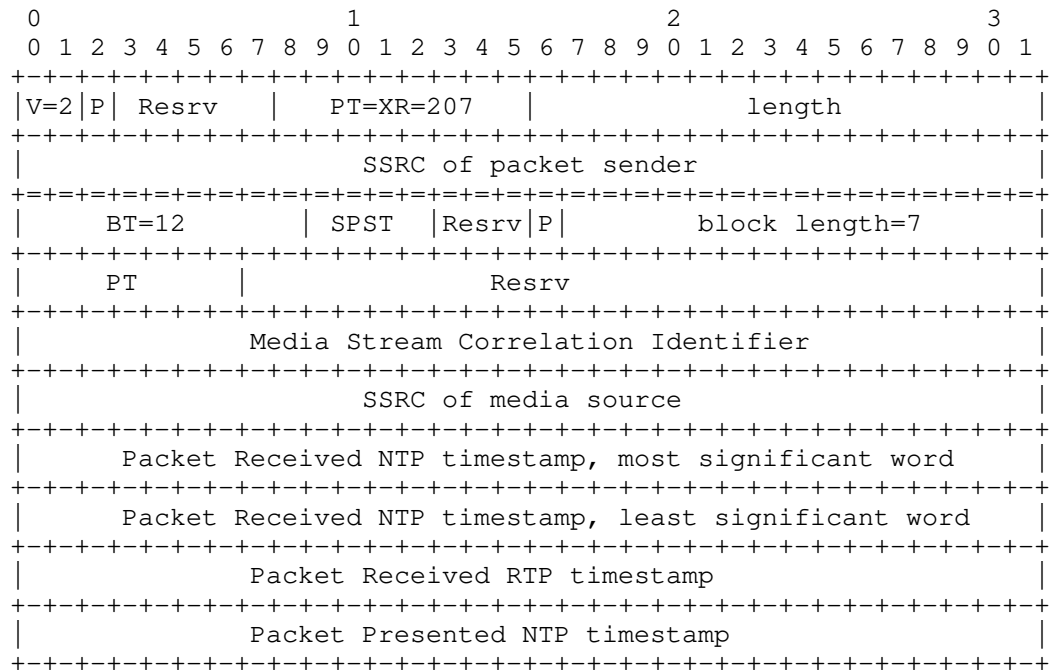
An SC reports RTP packet arrival times and play-out times of a media stream. It can receive summaries of such information, and use that to adjust its play-out buffer.

### 3.3. Mixer

A Mixer re-originates an RTP media stream. Therefore the SSRC and RTP time stamps of an outgoing RTP media stream of a mixer are unrelated to the SSRC and RTP time stamps of the incoming RTP stream. A Mixer can report the correlation between SSCRs and RTP time stamps of incoming and outgoing RTP media streams.

## 4. RTCP XR Block Type for IDMS

This section specifies the RTCP XR Block Type for reporting inter-destination media synchronization information on an RTP media stream. Its definition is based on [RFC5576]. The RTCP XR is used to report information on receipt times and presentation times of RTP packets to e.g. a Sender [RFC3611], a Feedback Target [RFC3550] or a Third Party Monitor [RFC3611]. The RTCP XR is also used to indicate synchronization settings instructions to a receiver of the RTP media stream.



The first 64 bits form the header of the RTCP XR, as defined in [RFC3611]. The SSRC of packet sender identifies the sender of the specific RTCP packet.

Block Type (BT): 8 bits. It identifies the block format. Its value shall be set to 12.

Synchronization Packet Sender Type (SPST): 4 bits. This field identifies the role of the packet sender for this specific eXtended Report. It can have the following values:

SPST=0 Reserved For future use.

SPST=1 The packet sender is an SC. It uses this XR to report synchronization status information. Timestamps relate to the SC input.

SPST=2 The packet sender is an MSAS. It uses this XR to report synchronization settings instructions. Timestamps relate to the input of a virtual SC, which acts as reference to which the SCs belonging to this session are synchronized.

SPST=3 The packet sender is a Mixer [RFC3550]. It uses this XR to report synchronization correlation information related to its incoming media stream. Timestamps relate to its input.

SPST=4 The packet sender is a Mixer [RFC3550]. It uses this XR to report synchronization correlation information related to a specific outgoing media stream of SC'. Timestamps relate to its input. (see Note below)

SPST=5-15 Reserved For future use.

NOTE: Following the RTP/RTCP specification [RFC3611], RTP timestamps relate to the arrival time of the first octet of an RTP packet. In case of SPST=4 (Mixer output), there is not such an arrival time as the media stream is re-originated at the Mixer. In this case, the timestamp would relate to the arrival time of the equivalent octet (representing e.g. the same video pixel or audio sample) of the incoming media stream.

Reserved bits (Resrv): 3 bits. These bits are reserved for future use and shall be set to 0.

Packet Presented NTP timestamp flag (P): 1 bit. Bit set to 1 if the Packet Presented NTP timestamp contains a value, 0 if it is empty. If this flag is set to zero, then the Packet Presented NTP timestamp shall not be inspected.

Block Length: 16 bits. This shall be set to 7, as this RTCP Block Type has a fixed length.

Payload Type (PT): 7 bits. This field identifies the format of the media payload, according to [RFC3551]. The media payload is associated with an RTP timestamp clock rate. This clock rate provides the time base for the RTP timestamp counter.

Reserved bits (Resrv): 25 bits. These bits are reserved for future use and shall be set to 0.

Media Stream Correlation Identifier: 32 bits. This identifier is used to correlate synchronized media streams. The value 0 (all bits are set "0") indicates that this field is empty. The value  $2^{32}-1$  (all bits are set "1") is reserved for future use. If the RTCP Packet Sender is an SC or an MSAS (SPST=1 or SPST=2), then the Media Stream Correlation Identifier maps on the SyncGroupId. If the RTCP Packet Sender is an Mixer (SPST=3 or SPST=4), related incoming and outgoing media streams have the same Media Stream Correlation Identifier.

SSRC: 32 bits. The SSRC of the media source shall be set to the value of the SSRC identifier carried in the RTP header [RFC3550] of the RTP packet to which the XR relates.

Packet Received NTP timestamp: 64 bits. This NTP timestamp [RFC5905] is the arrival wall clock time of the first octet of the RTP packet to which the XR relates. For SPST=2 it relates to a virtual SC to which the other SCs in the synchronization group may synchronize. For SPST=4 the SC' should calculate backwards when the content (video frame, audio sample) associated with the first octet of the RTP packet arrived. SCs shall be time-synchronized using e.g. NTP.

Packet Received RTP timestamp: 32 bits. This timestamp has the value of the RTP time stamp carried in the RTP header [RFC3550] of the RTP packet to which the XR relates.

Packet Presented NTP timestamp: 32 bits. This timestamp reflects the NTP time when the data contained in the first octet of the associated RTP packet is presented to the user. It consists of the least significant 16 bits of the NTP seconds part and the most significant 16 bits of the NTP fractional second part. If this field is empty, then it shall be set to 0 and the Packet Presented NTP timestamp flag (P) shall be set to 0. It shall be empty for SPST=3 and SPST=4.

## 5. SDP parameter for IDMS

The SDP parameter sync-group is used to signal the use of the RTCP XR block for inter-destination media synchronization. This SDP parameter extends rtcp-xr-attr as follows, using Augmented Backus-Naur Form [RFC5234].

```
rtcp-xr-attr = "a=" "rtcp-xr" ":" [xr-format *(SP xr-format)] CRLF
; Original definition from [RFC3611], section 5.1
```

```
xr-format = / grp-sync ; Extending xr-format for inter-destination
media synchronization
```

```
grp-sync = "grp-sync" [",sync-group=" SyncGroupId]
```

```
SyncGroupId = 1*DIGIT ; Numerical value from 0 till 4294967295
```

```
DIGIT = %x30-39
```

SyncGroupId is a 32-bit unsigned integer in network byte order and represented in decimal. SyncGroupId identifies a group of SCs for inter-destination media synchronization. It maps on the Media Stream Correlation Identifier of Annex W.1 for SPST=1 and SPST=2. The value

SyncGroupId=0 represents an empty SyncGroupId. The value 4294967295 ( $2^{32}-1$ ) is reserved for future use.

The following is an example of the SDP attribute for inter-destination media synchronization.

```
a=rtcp-xr:grp-sync,sync-group=42
```

## 6. Security Considerations

The specified RTCP XR Block Type in this document is used to collect, summarize and distribute information on packet-receipt times and play-out times of streaming media. The information may be used to orchestrate the media play-out at multiple devices.

Errors in the information, either accidental or malicious, may lead to undesired behavior. For example, if one device erroneously reports a two-hour delayed play-out, then another device in the same synchronization group could decide to delay its play-out by two hours as well, in order to keep its play-out synchronized. A user would likely interpret this two hour delay as a malfunctioning service.

Therefore, the application logic of both Synchronization Clients and Media Synchronization Application Servers should check for inconsistent information. Differences in play-out time of more than e.g. ten seconds could be an indication of such inconsistent information.

## 7. IANA Considerations

New block types for RTCP XR are subject to IANA registration. For general guidelines on IANA considerations for RTCP XR, refer to [RFC3611].

[TS 183 063] assigns the block type value 12 in the RTCP XR Block Type Registry to "Inter-destination Media Synchronization Block". [TS 183 063] also registers the SDP [RFC4566] parameter "grp-sync" for the "rtcp-xr" attribute in the RTCP XR SDP Parameters Registry.

## 8. Conclusions

This document specifies the RTCP XR and the associated SDP parameter for inter-destination media synchronization.

## 9. References

### 9.1. Normative References

- [RFC5234] Crocker, D. and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", RFC 5234, January 2008.
- [RFC3550] Schulzrinne, H., "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, July 2003.
- [RFC3551] Schulzrinne, H. and Casner S., "RTP Profile for Audio and Video Conferences with Minimal Control", RFC 3551, July 2003
- [RFC3611] Friedman, T. "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, November 2003.
- [RFC4566] Handley, M., "SDP: Session Description Protocol", RFC 4566, July 2006.
- [RFC5576] Lennox, J., "Source-Specific Media Attributes in the Session Description Protocol (SDP)", RFC 5576, June 2009.
- [RFC5905] Mills, D., "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.
- [TS 183 063] ETSI TISPAN, "IMS-based IPTV stage 3 specification", TS 183 063 v3.4.1, June 2010.

### 9.2. Informative References

- [Boronat2009] Boronat, F., et al, "Multimedia group and inter-stream synchronization techniques: A comparative study", Elsevier Information Systems 34 (2009), pp. 108-131

## 10. Acknowledgments

The authors thank TNO and KPN for each partially funding the work behind this internet draft.



Authors' Addresses

Hans M. Stokking  
TNO  
Brassersplein 2, Delft, the Netherlands  
  
Phone: +31 15 28 57078  
Email: hans.stokking@tno.nl

M. Oskar van Deventer  
TNO  
Brassersplein 2, Delft, the Netherlands  
  
Phone: +31 15 28 57078  
Email: oskar.vandeventer@tno.nl

Omar A. Niamut  
TNO  
Brassersplein 2, Delft, the Netherlands  
  
Phone: +31 15 28 57078  
Email: omar.niamut@tno.nl

Fabian A. Walraven  
TNO  
Brassersplein 2, Delft, the Netherlands  
  
Phone: +31 15 28 57078  
Email: fabian.walraven@tno.nl

Ray van Brandenburg  
TNO  
Brassersplein 2, Delft, the Netherlands  
  
Phone: +31 15 28 57078  
Email: ray.vanbrandenburg@tno.nl

Ishan Vaishnavi  
CWI  
Science Park 123, Amsterdam, the Netherlands  
  
Phone: +31 20 592 4323  
Email: i.vaishnavi@cwi.nl



AVT Working Group  
Internet Draft  
Intended status: Informational  
Expires: April 2011

G. Feher  
BME  
October 19, 2010

Using approximate authentication with Secure Real-time Transport  
Protocol (SRTP)  
draft-feher-avt-approx-auth-srtp-00.txt

#### 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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

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

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

This Internet-Draft will expire on April 19, 2011.

#### Copyright Notice

Copyright (c) 2010 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.

## Abstract

This document describes how to use an approximate authentication algorithm in the Secure Real-time Transport Protocol (SRTP) to provide integrity protection for the Real-time Transport Protocol (RTP) traffic.

Approximate authentication is a class of authentication algorithms where the authentication does not require the exact match of the source and received data, but a parameter given, certain amount of deviation is acceptable.

## Table of Contents

1. Introduction.....	2
2. Conventions used in this document.....	4
2.1. Stream ciphers and block ciphers.....	4
3. SRTP message authentication with approximate authentication....	4
3.1. SRTP headers and context.....	5
3.2. SRTP payload encryption.....	6
3.3. SRTP payload approximate authentication.....	6
3.4. Message authentication tag encryption.....	6
3.5. Key derivation for the algorithms.....	7
4. SRTCP message authentication.....	7
5. Security Considerations.....	8
6. IANA Considerations.....	8
7. References.....	8
7.1. Normative References.....	8
7.2. Informative References.....	9
8. Acknowledgments.....	9

## 1. Introduction

Message authentication is a protection tool to protect the transmitted data during the transmission. The protection can signal whether the transmitted message is the same at the sender and the receiver or not. Due to this technique no modifications by adversaries and no natural transmission errors remain undetectable by the receiver. Message authentication often utilizes keyed cryptographic hash functions.

In the case of wired data transmissions bit errors are rare, usually this channel is considered error free. In contrast, in the case of wireless channels bit errors during the transmission are common. The transmitted erroneous data frames are dropped by the receiver based on an integrity check. In order to avoid problems caused by these

drops, often an error correction protocol is applied at link level. The simplest error correction is to repeat the transmission when the receiver does not acknowledge the arrival of the transmitted data. This is used e.g. in WiFi unicast connections. However, there are certain scenarios where simple error correction is not applicable, and actually there is no error correction at all in this case. As an example, in the case of multicast WiFi transmissions, which are often used to transmit audio and video data, there is no protection against drops caused by natural bit errors.

Nowadays there are already available technologies that are able to cope with bit errors. Even if bit error correction is not possible, errors can be concealed. Depending on the amount of errors, the experienced results of bit error concealments could be better than dropping whole packets. Bit errors are even supported on transport layers with protocols, such as UDP-Lite [RFC3828]. This latter document also mentions radio technologies (e.g. [3GPP]) that permit partially damaged frames in the MAC layer.

Overall, there are real scenarios, where audio and video data are transmitted during error prone radio channels. Due to the error resilience and error concealment techniques in the decoders, users may benefit from the received, but not dropped damaged packets. However, when the user selects secure audio or video transmission, using SRTP [RFC3711] with a cryptographic hash based authentication, all the damaged packets should be dropped as they fail the authentication. For this reason, flows protected with current SRTP algorithms are not able to profit from the damaged packets.

Approximate authentication is a class of authentication algorithms where the authentication does not require the exact match of the source and received data [DONGVU][FEHER][GRAVISH]. The user is able to give a threshold for the number of mismatching bits, and if the number of errors is below of such threshold, then the packet is authenticated. Authenticating a modified packet is not necessarily a security problem. When the authentication threshold is low, then the adversary is not able to spoof the whole content. Assuming that the content is encrypted, the adversary cannot perform predictable content spoofing. No hijacking is possible either. As a maximum, the adversary may turn down the quality of the transmitted media by modifying some bits in the stream, but this impact should be small compared to the case, where damaged packets are dropped.

## 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 RFC-2119 [RFC2119].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

### 2.1. Stream ciphers and block ciphers

In this draft ciphers are mentioned at several places. Basically ciphers could be block ciphers or stream ciphers. The differences between the two types of ciphers are:

- o Block ciphers deal with larger blocks of information (i.e. usually 64 or 128 bit blocks), while stream ciphers work bit by bit.
- o Stream ciphers have memory

It is possible to convert block ciphers to stream ciphers back and forth, so it might not be obvious when a certain cipher in a certain operation mode is a block cipher or a stream cipher. Further on in this document, the stream ciphers refer to cryptographic algorithms that use some function (i.e. mostly the XOR function) to combine a pseudorandom number generator (PRNG) with a plaintext/ciphertext stream. They are also called as combiner-type algorithms in some NSA documents. Block ciphers refer to algorithms that operate on groups of bits of a fixed length, termed blocks. An important property of a block cipher is the avalanche effect, in the meaning that a slight change in the ciphertext causes a significant change in the decoded plaintext.

According to the previous distinction, AES-ICM (i.e. AES block cipher in Integer Counter Mode) is considered as a stream cipher, while AES-CBC (i.e. AES block cipher in Cipher-block Chaining mode) is considered as block cipher.

## 3. SRTP message authentication with approximate authentication

During the approximate authentication procedure, at the source side, a message authentication tag will be calculated and inserted to the authentication part of the SRTP message. This authentication takes care of the message header and payload separately. For security reason the message authentication tag is encrypted.

At the receiver side the included message authentication tag is decrypted and compared to the code that is calculated over the received message, same way as it was on the sender side. When the difference is below the given threshold, then the message is authenticated, otherwise the error audit message "AUTHENTICATION FAILURE" MUST be returned.

The calculation of the message authentication tag:

Authentication tag =

$$E_{k1}(H_{k2}(SRTP\ header || ROC) + AA_{k3}(SRTP\ payload))),$$

where E is an encryption algorithm using a block cipher, H is a cryptographic hash algorithm and AA is the approximate authentication algorithm. Their keys are k1, k2 and k3 respectively. These subkeys are generated using the negotiated authentication key.

The acceptance threshold parameter extends SRTP context. The value of the parameter SHOULD be decided during the negotiation of other parameters.

### 3.1. SRTP headers and context

In the SRTP message the authentication covers the header and the payload. The header section of the SRTP packet is considered as highly error sensitive. Furthermore, a possible attack against sequence number and other fields makes the header sensitive from security point of view as well. Thus, the SRTP header part requires perfect authentication, so it MUST be authenticated using a cryptographic hash function (e.g. HMAC-SHA1 [RFC2104]).

To perform the header authentication, a cryptographic hash function MUST be used. When the size of the message authentication tag (n\_tag) is shorter than the output of the hash function, then only the leftmost n\_tag bit SHOULD be considered. Otherwise, when the message authentication tag is longer than the hash output, then the hash value SHOULD be repeated along the length of the message authentication tag.

According to RFC3711, in the case of SRTP, the authentication should cover the ROC value from the cryptographic context as well. Since ROC is a sensitive data in terms of security, it requires perfect authentication. For this reason the ROC is concatenated to the SRTP header and the cryptographic hash function covers them both.

### 3.2. SRTP payload encryption

The approximate authentication of the message allows the successful authentication of the packet regardless a small number of bit errors in the payload. In order to circumvent content spoofing attacks, it is RECOMMENDED to encrypt the content. When the content is encrypted, the encryption algorithm MUST be a stream cipher. In other cases, due to the significant change caused by the avalanche effect, using approximate authentication is meaningless.

Since the encrypted payload use a stream cipher for the encryption procedure, there is no need for padding. Due to the nature of approximate authentication, as padding length can be spoofed, padding is risky. In fact, RTP padding MUST be disabled, so the packet SHOULD NOT contain any RTP padding or RTP pad count fields.

### 3.3. SRTP payload approximate authentication

The approximate authentication algorithm creates a checksum like authentication code. Using the code of the transmitted payload and the code calculated at the sender side, it is possible to deduce to the amount of modifications in the payload. The approximate authentication is not necessarily an error detection code, therefore it may happen that the exact number of differences cannot be signaled. In this situation the code gives an approximate value only, hence the name is approximate authentication.

Calculating the approximate authentication code MAY require the need of ROC and SEQ from the cryptographic context. Together with these values and the key associated to the approximate authentication, it is possible to perform packet specific cryptographic operations safely.

### 3.4. Message authentication tag encryption

The encryption of the message authentication tag is necessary, since the approximate authentication field might be subject of spoofing attacks. The encryption procedure can provide protection against spoofing. Stream ciphers are inappropriate for this protection, since this way the attacker can perform predictable changes in the decrypted authentication tag. The encryption algorithm used here MUST be a block cipher.

It is possible that the length of the message authentication tag is not the multiple of the block size, which the block cipher defines. Block ciphers having a block size larger than the message authentication tag length MUST NOT be used. The output of the block



cipher cannot be cut, just like hash outputs. Otherwise, there are two options. When the length of the message authentication tag is exactly the same as the block size, then any block cipher can be used. When the length of the message authentication tag is greater than the block size, then the Ciphertext stealing (CTS) operation mode [CTS] MUST be used. Using the CTS method, the size of the message authentication tag should not be aligned to the block size.

For short message authentication tags, it is RECOMMENDED to select a block cipher that has 64 bit block size (e.g. BLOWFISH [BF]).

During the transmission the message authentication tag may suffer bit error damage. Due to the avalanche effect in block ciphers, the authentication will fail in this case.

### 3.5. Key derivation for the algorithms

The encryption algorithm, used to encrypt the message authentication tag, the cryptographic hash function, used to protect the SRTP header and the approximate authentication algorithm, used to protect the SRTP payload require keys. The encryption algorithm MUST have a key, while at the hash function and at the approximate authentication, keys are OPTIONAL only. Despite these algorithms may work without a key, keyed algorithms are RECOMMENDED increasing the security. These three keys are called subkeys and managed by algorithms described in SRTP.

The subkeys MUST be generated using the authentication key (the `k_a` parameter in RFC3711) via the key derivation method defined in RFC3711. The key derivation MUST use the AES-CM PRF, which is mandatory to implement in SRTP. There are new labels defined for the new keys. For the encryption key the `<label> = 0x06`, for the hash function `<label> = 0x07` and for the approximate authentication `<label> = 0x08`.

The length of the subkeys should be defined together with the particular approximate authentication algorithm.

### 4. SRTCP message authentication

The approximate authentication is valid only for SRTP messages. SRTCP SHOULD NOT use approximate authentication. In the SRTCP messages there are no data that could be used with bit damages. For this reason approximate authentication is not applicable.

SRTCP is out of scope of this document.

## 5. Security Considerations

The security considerations in RFC3711 apply to this document as well.

Section 9.5 of RFC3711 considers weak authentication in terms of security. The approximate authentication can be considered as a weak authentication, since due to its nature, it authenticates modified messages. It is impossible to get know whether the source of the modification is an adversary or it is a natural error.

SRTP MAY be used with weak authentication, where it is an acceptable security risk, and it is impractical to provide strong message authentication. The risks associated with exercising the weak authentication need to be considered by a security audit prior to its use for a particular application or environment. Further details can be found in RFC3711.

## 6. IANA Considerations

SRTP uses cryptographic transforms which a key management protocol signals. It is the task of each such protocol to register the cryptographic transforms or suites of transforms with IANA. This draft defines no new cryptographic transforms or suites of transforms, therefore IANA considerations can be omitted.

IANA will register new crypto suites into the subregistry for SRTP crypto suites, when a particular approximate authentication algorithm will be proposed.

## 7. References

### 7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2234] Crocker, D. and Overell, P. (Editors), "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, Internet Mail Consortium and Demon Internet Ltd., November 1997.
- [CTS] Schneier, Bruce, "Applied Cryptography", Second Edition, John Wiley and Sons, New York, 1996. Errata: on page 195, line 13, the reference number should be [402].

## 7.2. Informative References

- [RFC2104] Krawczyk, H., Bellare, M. and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.
- [RFC3828] Larzon, L-A., Degermark, M., Pink, S., Jonsson, L-E., and G. Fairhurst, "The Lightweight User Datagram Protocol (UDP-Lite)", RFC 3828, July 2004.
- [3GPP] "Technical Specification Group Services and System Aspects; Quality of Service (QoS) concept and architecture", TS 23.107 V5.9.0, Technical Specification 3rd Generation Partnership Project, June 2003.
- [BF] B. Schneier, "Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish)", Fast Software Encryption, Cambridge Security Workshop Proceedings (December 1993), Springer-Verlag, 1994, pp. 191-204.
- [DONGVU] Dongvu Tonien, Reihaneh Safavi-Naini, Peter Nickolas, Yvo Desmedt, "Unconditionally Secure Approximate Message Authentication", IWCC, 2009, pp. 233-247
- [FEHER] Gabor Feher, Istvan Olah, "Enhancing wireless video streaming using lightweight approximate authentication", Multimedia Systems Journal (MMS), 2008, Vol 14(3), pp. 167-177
- [GRAVISH] Liehua Xie, Gonzalo R. Arce, R. F. Graveman, "Approximate image message authentication codes", IEEE Transactions on Multimedia (TMM), 2001, Vol. 3(2), pp. 242-252

## 8. Acknowledgments

This draft is based on the experience gained in the OPTIMIX research project. The research leading to these results has received funding from the European Union's Seventh Framework Programme ([FP7/2007-2013]) under grant agreement no ICT-214625.

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Gabor Feher  
Budapest University of Technology and Economics  
H-1117 Budapest, Magyar tudosok krt. 2., HUNGARY  
Email: feher10@tmit.bme.hu



Audio/Video Transport Working  
Group  
Internet-Draft  
Intended status: Informational  
Expires: November 19, 2010

G. Hunt  
P. Arden  
BT  
May 18, 2010

Monitoring Architectures for RTP  
draft-hunt-avt-monarch-01.txt

Abstract

This memo proposes an architecture for extending RTCP with a new RTCP XR (RFC3611) block type to report new metrics regarding media transmission or reception quality, as proposed in draft-ietf-avt-rtcp-guidelines (work in progress [replace with RFC number]). This memo suggests that a new block should contain a single metric or a small number of metrics relevant to a single parameter of interest or concern, rather than containing a number of metrics which attempt to provide full coverage of all those parameters of concern to a specific application. Applications may then "mix and match" to create a set of blocks which covers their set of concerns. Where possible, a specific block should be designed to be re-usable across more than one application, for example, for all of voice, streaming audio and video.

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 November 19, 2010.

Copyright Notice

Copyright (c) 2010 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 . . . . .	3
2. Requirements notation . . . . .	4
3. Using small blocks . . . . .	5
4. The identity block . . . . .	6
5. An example of a metric block . . . . .	10
6. Application to translators . . . . .	11
7. Application to RFC 5117 topologies . . . . .	12
8. Expanding the RTCP XR block namespace . . . . .	13
9. IANA Considerations . . . . .	14
10. Security Considerations . . . . .	15
11. Informative References . . . . .	16
Authors' Addresses . . . . .	17

## 1. Introduction

Any proliferation of metrics for transport and application quality monitoring has been identified as a potential problem for RTP/RTCP interoperability. Different applications layered on RTP may have some monitoring requirements in common, which should be satisfied by a common design. The objective here is to define an extensible framework and a small number of re-usable metrics to reduce implementation costs and to maximise inter-operability. Work-in-progress on [GUIDELINES] has stated that, where RTCP is to be extended with a new metric, the preferred mechanism is by the addition of a new RTCP XR [RFC3611] block. This memo assumes that any requirement for a new metric to be transported in RTCP will use a new RTCP XR block.

[GUIDELINES] provides advice on when and how new metrics should be introduced, including recommending that metrics are based on existing standards whenever possible.

Section 3 describes the key proposal of thismemo, the use of small metrics blocks each of which addresses a single parameter of interest which may be "mixed and matched", rather than providing a large block to address all the parameters which might be of interest to a broad class of applications (for example, all VoIP applications).

Section 4 describes an optimisation to avoid repetition of identification information, which becomes desirable when small blocks are used.

Section 5 provides an example of the application of these principles to a specific case, that of a metric block to report packet delay variation.

Section 6 draws attention to the guidance in [RFC3550] concerning RTCP and translators.

Section 7 discusses the potential application of RTCP XR metrics blocks to the conferencing topologies discussed in [RFC5117].

Section 8 consists (in this draft) only of an "Editor's note" asking whether the limited namespace available for RTCP XR blocks is a concern, and if so whether it would be desirable to work on a standardised means to expand it.



## 2. Requirements notation

This memo is informative and as such contains no normative requirements.

### 3. Using small blocks

Different applications using RTP for media transport certainly have differing requirements for metrics transported in RTCP to support their operation. For many applications, the basic metrics for transport impairments provided in RTCP SR and RR packets [RFC3550] (together with source identification provided in RTCP SDES packets) are sufficient. For other applications additional metrics may be required or at least sufficiently useful to justify the overheads, both of processing in endpoints and of increased session bandwidth. For example an IPTV application using Forward Error Correction (FEC) might use either a metric of post-repair loss or a metric giving detailed information about pre-repair loss bursts to optimise payload bandwidth and the strength of FEC required for changing network conditions. However there are many metrics available. It is likely that different applications or classes of applications will wish to use different metrics. Any one application is likely to require metrics for more than one parameter but if this is the case, different applications will almost certainly require different combinations of metrics. If larger blocks are defined containing multiple metrics to address the needs of each application, it becomes likely that many different such larger blocks are defined, which becomes a danger to interoperability.

To avoid this pitfall, this memo proposes the use of small RTCP XR metrics blocks each containing a very small number of individual metrics characterising only one parameter of interest to an application running over RTP. For example, at the RTP transport layer, the parameter of interest might be packet delay variation, and specifically the metric "IPDV" defined by [Y1540]. See Section 5 for architectural considerations for a metrics block, using as an example a metrics block to report packet delay variation.

#### 4. The identity block

Any measurement must be identified. However if metrics are delivered in small blocks there is a danger of inefficiency arising from repeating this information in a number of metrics blocks within the same RTCP packet, in cases where the same identification information applies to multiple metrics blocks.

An instance of a metric must be identified using information which is likely to include most of the following:

- o the node at which it was measured,
- o the source of the measured stream (for example, its CNAME),
- o the SSRC of the measured stream,
- o the sequence number of the first packet of the RTP session,
- o the extended sequence numbers of the first packet of the current measurement interval, and the last packet included in the measurement,
- o the duration of the most recent measurement interval and
- o the duration of the interval applicable to cumulative measurements (which may be the duration of the RTP session to date).

[Editor's note: this set of information overlaps with, but is more extensive than, that in the union of similar information in RTCP RR packets. Should we assume that RR information is always present if XR is sent, and that measurement intervals are exactly coincident? If so, state assumption and remove overlaps. What were the design considerations which led to the additional information \*not\* being present in RRs? The reason for the additional information here is the perceived difficulty of "locating" the \*start\* of the RTP session (sequence number of 1st packet, duration of interval applicable to cumulative measurements) using only RR. Is this a misconception? It leads to redundant information in this design because equivalent information is provided multiple times, once in \*every\* identification packet. Though this ensures immunity to packet loss, the design is ugly and the overhead is not completely trivial.]

This section proposes an approach to minimise the inefficiency of providing this identification information, assuming that an architecture based on small blocks means that a typical RTCP packet will contain more than one metrics block needing the same identification. The choice of identification information to be

provided is discussed in [IDENTITY] (work in progress).

The approach is to define a stand-alone block containing only identification information, and to tag this identification block with a number which is unique within the scope of the containing RTCP XR packet. The "containing RTCP XR packet" is defined here as the RTCP XR header with PT=XR=207 defined in Section 2 of [RFC3611] and the associated payload defined by the length field of this RTCP XR header. The RTCP XR header itself includes the SSRC of the node at which all of the contained metrics were measured, hence this SSRC need not be repeated in the stand-alone identification block. A single containing RTCP XR packet may contain multiple identification blocks limited by the range of the tag field. Typically there will be one identification block per monitored source SSRC, but the use of more than one identification block for a single monitored source SSRC within a single containing RTCP XR packet is not ruled out.

There will be zero or more metrics blocks dependent on each identification block. The dependence of an instance of a metrics block on an identification block is established by the metrics block's having the same numeric value of the tag field as its identification block (in the same containing RTCP XR packet).

Figure 1 below illustrates this principle using as an example an RTCP XR packet containing four metrics blocks, reporting on streams from two sources. The measurement identity information is provided in two blocks with Block Type NMI, and tag values 0 and 1 respectively.

Note: in this example, RTCP XR block type values for four proposed new block types (work in progress) are given as NMI, NPDV, NBGL and NDEL. These represent numeric block type codepoints to be allocated by IANA at the conclusion of the work.

Each of these two identity blocks will specify the SSRC of one of the monitored streams, as well as information about the span of the measurement. There are two metrics blocks with tag=0 indicating their association with the measurement identity block which also has tag=0. These are the two blocks following the identity block with tag=0, though this positioning is not mandatory. There are also two metrics blocks with tag=1 indicating their association with the measurement identity block which also has tag=1, and these are the two blocks following the identity block with tag=1.

[Editor's note: if we mandated that metrics blocks associated with an identity block must always follow the identity block we could save the tag field and possibly simplify processing. Is this preferable to cross-referencing with a numeric tag?]

In the example, the block types of the metrics blocks associated with tag=0 are BT=NPDV (a PDV metrics block) and BT=NBGL (a burst and gap loss metrics block). The block types of the metrics blocks associated with tag=1 are BT=NPDV (a second PDV metrics block) and BT=NDEL (a delay metrics block). This illustrates that:

- o multiple instances of the same metrics block may occur within a containing RTCP XR packet, associated with different identification information, and
- o differing measurements may be made, and reported, for the different streams arriving at an RTP system.

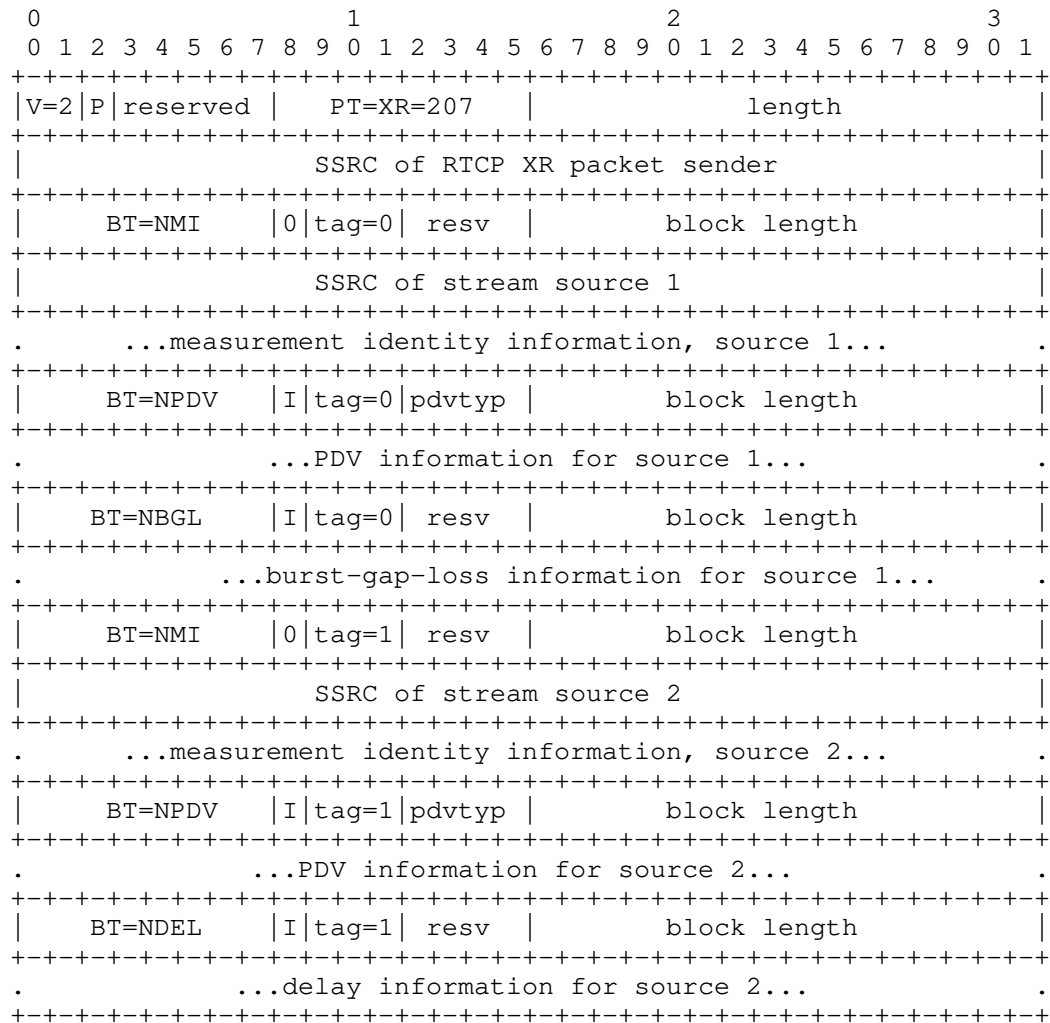


Figure 1: RTCP XR block with identity blocks

This approach of separating the identification information is more costly than providing identification in each metrics block if only a single metrics block is sent in an RTCP packet, but becomes beneficial as soon as more than one metrics block shares common identification.

## 5. An example of a metric block

This section uses the example of an existing proposed metrics block to illustrate the application of the principles set out in Section 3.

The example [PDV] (work in progress) is a block to convey information about packet delay variation (PDV) only, consistent with the principle that a metrics block should address only one parameter of interest. One simple metric of PDV is available in the RTCP RR packet as the "jit" field. There are other PDV metrics which may be more useful to certain applications. Two such metrics are the IPDV metric ([Y1540], [RFC3393]) and the MAPDV2 metric [G1020]. Use of these metrics is consistent with the principle in Section 5 of [GUIDELINES] that metrics should usually be defined elsewhere, so that RTCP standards define only the transport of the metric rather than its nature. The purpose of this section is to illustrate the architecture using the example of [PDV] (work in progress) rather than to document the design of the PDV metrics block or to provide a tutorial on PDV in general.

Given the availability of at least three metrics for PDV, there are design options for the allocation of metrics to RTCP XR blocks:

- o provide an RTCP XR block per metric
- o provide a single RTCP XR block which contains all three metrics
- o provide a single RTCP block to convey any one of the three metrics, together with a identifier to inform the receiving RTP system of the specific metric being conveyed

In choosing between these options, extensibility is important, because additional metrics of PDV may well be standardised and require inclusion in this framework. The first option is extensible but only by use of additional RTCP XR blocks, which may consume the limited namespace for RTCP XR blocks at an unacceptable rate. The second option is not extensible, so could be rejected on that basis, but in any case a single application is quite unlikely to require transport of more than one metric for PDV. Hence the third option was chosen. This implies the creation of a subsidiary namespace to enumerate the PDV metrics which may be transported by this block, as discussed further in [PDV] (work in progress).

## 6. Application to translators

Section 7.2 of [RFC3550] describes processing of RTCP by translators. RTCP XR is within the scope of the recommendations of [RFC3550]. Some RTCP XR metrics blocks may usefully be measured at, and reported by, translators. As described in [RFC3550] this creates a requirement for the translator to allocate an SSRC for itself so that it may populate the SSRC in the RTCP XR packet header (although the translator is not a Synchronisation Source in the sense of originating RTP media packets). It must also supply this SSRC and the corresponding CNAME in RTCP SDES packets.

In RTP sessions where one or more translators generate any RTCP traffic towards their next-neighbour RTP system, other translators in the session have a choice as to whether they forward a translator's RTCP packets. Forwarding may provide additional information to other RTP systems in the connection but increases RTCP bandwidth and may in some cases present a security risk. RTP translators may have forwarding behaviour based on local policy, which might differ between different interfaces of the same translator.

[Editor's note: for bidirectional unicast, an RTP system may usually detect RTCP from a translator by noting that the sending SSRC is not present in any RTP media packet. However even for bidirectional unicast there is a possibility of a source sending RTCP before it has sent any RTP media (leading to transient mis-categorisation of an RTP end system or RTP mixer as a translator), and for multicast sessions - or unidirectional/streaming unicast - there is a possibility of a receive-only end system being permanently mis-categorised as a translator. Is there a need for a translator to declare itself explicitly? Needs further thought.]



## 7. Application to RFC 5117 topologies

An RTP system (end system, mixer or translator) which originates, terminates or forwards RTCP XR blocks is expected to handle RTCP, including RTCP XR, as specified in [RFC3550] for that class of RTP systems. Provided this expectation is met, an RTP system using RTCP XR is architecturally no different from an RTP system of the same class (end system, mixer, or translator) which does not use RTCP XR. This statement applies to the topologies investigated in [RFC5117], where they use RTP end systems, RTP mixers and RTP translators as these classes are defined in [RFC3550].

These topologies are specifically Topo-Point-to-Point, Topo-Multicast, Topo-Translator (both variants, Topo-Trn-Translator and Topo-Media-Translator, and combinations of the two), and Topo-Mixer.

The topologies based on systems which do not behave according to [RFC3550], that is Topo-Video-Switch-MCU and Topo-RTCP-terminating-MCU, suffer from the difficulties described in [RFC5117]. These difficulties apply to systems sending, and expecting to receive, RTCP XR blocks as much as to systems using other RTCP packet types. For example, a participant RTP end system may send media to a video switch MCU. If the media stream is not selected for forwarding by the switch, neither RTCP RR packets nor RTCP XR blocks referring to the end system's generated stream will be received at the RTP end system. Strictly the RTP end system can only conclude that its RTP has been lost in the network, though an RTP end system complying with the robustness principle of [RFC1122] should survive with essential functions unimpaired.

## 8. Expanding the RTCP XR block namespace

[Editor's note: the RTCP XR block namespace is limited by the 8-bit block type field in the RTCP XR header (Section 3 of [RFC3611]). IESG have noted that this is potentially restrictive. It would be possible to standardise an expansion mechanism, probably based on use of a new field near the start of the variable-length "type-specific block contents" field. Clearly this could apply only to new block types, so might be standardised to apply to some subrange of the current 8-bit range, for example the range 128 through 191 might be used. At time of writing, block types 12 to 254 are unassigned and 255 is reserved for future expansion. Is there a consensus for, or against, work to allow expansion? One potential use is through hierarchical control, where one or a few codepoints at the top level are given to other SDOs who may then define a number of metrics distinguished by values in the (so far hypothetical) new field.]

## 9. IANA Considerations

None.

## 10. Security Considerations

This document itself contains no normative text and hence should not give rise to any new security considerations, to be confirmed.

## 11. Informative References

- [G1020] ITU-T, "ITU-T Rec. G.1020, Performance parameter definitions for quality of speech and other voiceband applications utilizing IP networks", July 2006.
- [GUIDELINES] Ott, J., "Guidelines for Extending the RTP Control Protocol (RTCP)", ID draft-ott-avt-rtcp-guidelines-03, February 2010.
- [IDENTITY] Hunt, G., "RTCP XR Report Block for Measurement Identity", ID draft-ietf-avt-rtcp-xr-meas-identity-02, May 2009.
- [PDV] Hunt, G., "RTCP XR Report Block for Packet Delay Variation Metric Reporting", ID draft-ietf-avt-rtcp-xr-pdv-03, May 2009.
- [RFC1122] Braden, R., "Requirements for Internet Hosts -- Communication Layers", RFC 1122, October 1989.
- [RFC3393] Demichelis, C., "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC 3393, November 2002.
- [RFC3550] Schulzrinne, H., "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, July 2003.
- [RFC3611] Friedman, T., "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, November 2003.
- [RFC5117] Westerlund, M., "RTP Topologies", RFC 5117, January 2008.
- [Y1540] ITU-T, "ITU-T Rec. Y.1540, IP packet transfer and availability performance parameters", November 2007.

Authors' Addresses

Geoff Hunt  
BT  
Orion 1 PP2  
Adastral Park  
Martlesham Heath  
Ipswich, Suffolk IP5 3RE  
United Kingdom

Phone: +44 1473 651704  
Email: geoff.hunt@bt.com

Philip Arden  
BT  
Orion 3/7 PP4  
Adastral Park  
Martlesham Heath  
Ipswich, Suffolk IP5 3RE  
United Kingdom

Phone: +44 1473 644192  
Email: philip.arden@bt.com



Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: April 28, 2011

M. Westerlund  
I. Johansson  
Ericsson  
C. Perkins  
University of Glasgow  
P. O'Hanlon  
UCL  
K. Carlberg  
G11  
October 25, 2010

Explicit Congestion Notification (ECN) for RTP over UDP  
draft-ietf-avt-ecn-for-rtp-03

Abstract

This document specifies how explicit congestion notification (ECN) can be used with RTP/UDP flows that use RTCP as feedback mechanism.

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 April 28, 2011.

Copyright Notice

Copyright (c) 2010 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 . . . . .	3
2. Conventions, Definitions and Acronyms . . . . .	4
3. Discussion, Requirements, and Design Rationale . . . . .	4
3.1. Requirements . . . . .	6
3.2. Applicability . . . . .	7
4. Overview of Use of ECN with RTP/UDP/IP . . . . .	10
5. RTCP Extensions for ECN feedback . . . . .	13
5.1. RTP/AVPF Transport Layer ECN Feedback packet . . . . .	13
5.2. RTCP XR Report block for ECN summary information . . . . .	16
6. Use of ECN with RTP/UDP/IP . . . . .	17
6.1. Negotiation of ECN Capability . . . . .	17
6.2. Initiation of ECN Use in an RTP Session . . . . .	21
6.3. Ongoing Use of ECN Within an RTP Session . . . . .	27
6.4. Detecting Failures . . . . .	29
7. Processing RTCP ECN Feedback in RTP Translators and Mixers . . . . .	33
7.1. Fragmentation and Reassembly in Translators . . . . .	33
7.2. Generating RTCP ECN Feedback in Media Transcoders . . . . .	35
7.3. Generating RTCP ECN Feedback in Mixers . . . . .	36
8. Implementation considerations . . . . .	36
9. IANA Considerations . . . . .	36
9.1. SDP Attribute Registration . . . . .	37
9.2. RTP/AVPF Transport Layer Feedback Message . . . . .	37
9.3. RTCP XR Report blocks . . . . .	37
9.4. STUN attribute . . . . .	37
9.5. ICE Option . . . . .	38
10. Security Considerations . . . . .	38
11. Examples of SDP Signalling . . . . .	40
12. Open Issues . . . . .	40
13. References . . . . .	41
13.1. Normative References . . . . .	41
13.2. Informative References . . . . .	42
Authors' Addresses . . . . .	43

## 1. Introduction

This document outlines how Explicit Congestion Notification (ECN) [RFC3168] can be used for RTP [RFC3550] flows running over UDP/IP which use RTCP as a feedback mechanism. The solution consists of feedback of ECN congestion experienced markings to the sender using RTCP, verification of ECN functionality end-to-end, and how to initiate ECN usage. The initiation process will have some dependencies on the signalling mechanism used to establish the RTP session, a specification for signalling mechanisms using SDP is included.

ECN is getting attention as a method to minimise the impact of congestion on real-time multimedia traffic. When ECN is used, the network can signal to applications that congestion is occurring, whether that congestion is due to queuing at a congested link, limited resources and coverage on a radio link, or other reasons.

ECN provides a way for networks to send congestion control signals to a media transport without having to impair the media. Unlike losses, the signals unambiguously indicate congestion to the transport as quickly as feedback delays allow, and without confusing congestion with losses that might have occurred for other reasons such as transmission errors, packet-size errors, routing errors, badly implemented middleboxes, policy violations and so forth.

The introduction of ECN into the Internet requires changes to both the network and transport layers. At the network layer, IP forwarding has to be updated to allow routers to mark packets, rather than discarding them in times of congestion [RFC3168]. In addition, transport protocols have to be modified to inform the sender that ECN marked packets are being received, so it can respond to the congestion. TCP [RFC3168], SCTP [RFC4960] and DCCP [RFC4340] have been updated to support ECN, but to date there is no specification how UDP-based transports, such as RTP [RFC3550], can use ECN. This is due to the lack of feedback mechanisms directly in UDP. Instead the signaling control protocol on top of UDP needs to provide that feedback, which for RTP is RTCP.

The remainder of this memo is structured as follows. We start by describing the conventions, definitions and acronyms used in this memo in Section 2, and the design rationale and applicability in Section 3. Section 4 provides an overview of how ECN is used with RTP over UDP. Then the definition of the RTCP extensions for ECN feedback in Section 5. Then the full details of how ECN is used with RTP over UDP is defined in Section 6. In Section 7 we discuss how RTCP ECN feedback is handled in RTP translators and mixers. Section 8 discusses some implementation considerations, Section 9

lists IANA considerations, and Section 10 discusses the security considerations.

## 2. Conventions, Definitions and Acronyms

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].

### Abbreviations

- o ECN: Explicit Congestion Notification
- o ECT: ECN Capable Transport
- o ECN-CE: ECN Congestion Experienced
- o not-ECT: Not ECN Capable Transport

The meaning of the term ECN support depends on which entity between the sender and receiver (inclusive) that is considered. We distinguish between:

- o ECN-Capable Host: Sender or receiver of media.
- o ECN-Capable Transport: ECT = all ends are ECN capable hosts.
- o ECN-Capable Packets: Packets are either ECT or CE.
- o ECN-Oblivious Relay: Router or middlebox that treats ECN-Capable Packets no differently from Not-ECT.
- o ECN-Capable Queue: Supports ECN marking of ECN-Capable Packets.
- o ECN-Blocking Middlebox: Discards ECN-Capable Packets.
- o ECN-Reverting Middlebox: Changes ECN-Capable Packets to Not-ECT.

## 3. Discussion, Requirements, and Design Rationale

ECN has been specified for use with TCP [RFC3168], SCTP [RFC4960], and DCCP [RFC4340] transports. These are all unicast protocols which negotiate the use of ECN during the initial connection establishment handshake (supporting incremental deployment, and checking if ECN marked packets pass all middleboxes on the path). ECN Congestion Experienced (ECN-CE) marks are immediately echoed back to the sender

by the receiving end-point using an additional bit in feedback messages, and the sender then interprets the mark as equivalent to a packet loss for congestion control purposes.

If RTP is run over TCP, SCTP, or DCCP, it can use the native ECN support provided by those protocols. This memo does not concern itself further with these use cases. However, RTP is more commonly run over UDP. This combination does not currently support ECN, and we observe that it has significant differences from the other transport protocols for which ECN has been specified. These include:

**Signalling:** RTP relies on separate signalling protocols to negotiate parameters before a session can be created, and doesn't include an in-band handshake or negotiation at session set-up time (i.e. there is no equivalent to the TCP three-way handshake in RTP).

**Feedback:** RTP does not explicitly acknowledge receipt of datagrams. Instead, the RTP Control Protocol (RTCP) provides reception quality feedback, and other back channel communication, for RTP sessions. The feedback interval is generally on the order of seconds, rather than once per network RTT (although the RTP/AVPF profile [RFC4585] allows more rapid feedback in most cases).

**Congestion Response:** While it is possible to adapt the transmission of many audio/visual streams in response to network congestion, and such adaptation is required by [RFC3550], the dynamics of the congestion response may be quite different to those of TCP or other transport protocols.

**Middleboxes:** The RTP framework explicitly supports the concept of mixers and translators, which are middleboxes that are involved in media transport functions.

**Multicast:** RTP is explicitly a group communication protocol, and was designed from the start to support IP multicast (primarily ASM, although a recent extension supports SSM with unicast feedback [RFC5760]).

**Application Awareness:** ECN support via TCP, DCCP, and SCTP constrain the awareness and reaction to packet loss within those protocols. By adding support of ECN through RTCP, the application is made aware of packet loss and may choose one or more approaches in response to that loss.

**Counting vs Detecting Congestion:** TCP and the protocols derived from it are mainly designed to respond the same whether they experience a burst of congestion indications within one RTT or just one. Whereas real-time applications may be concerned with the amount of

congestion experienced, whether it is distributed smoothly or in bursts. When feedback of ECN was added to TCP [RFC3168], the receiver was designed to flip the echo congestion experienced (ECE) flag to 1 for a whole RTT then flop it back to zero. Whereas ECN feedback in RTCP will need to report a count of how much congestion has been experienced within an RTCP reporting period, irrespective of round trip times.

These differences will significantly alter the shape of ECN support in RTP-over-UDP compared to ECN support in TCP, SCTP, and DCCP, but do not invalidate the need for ECN support.

ECN support is more important for RTP sessions than for instance is the case for TCP- This because the impact of packet loss in real-time audio-visual media flows is highly visible to users. Effective ECN support for RTP flows running over UDP will allow real-time audio-visual applications to respond to the onset of congestion before routers are forced to drop packets, allowing those applications to control how they reduce their transmission rate, and hence media quality, rather than responding to, and trying to conceal the effects of, unpredictable packet loss. Furthermore, widespread deployment for ECN and active queue management in routers, should it occur, can potentially reduce unnecessary queueing delays in routers, lowering the round-trip time and benefiting interactive applications of RTP, such as voice telephony.

### 3.1. Requirements

Considering ECN, transport protocols supporting ECN, and RTP based applications one can create a set of requirements that must be satisfied to at least some degree if ECN is to be used by RTP over UDP.

- o REQ 1: A mechanism MUST negotiate and initiate the usage of ECN for RTP/UDP/IP sessions so that an RTP sender will not send packets with ECT in the IP header unless it knows all potential receivers will understand any CE indications they might receive.
- o REQ 2: A mechanism MUST feedback the reception of any packets that are ECN-CE marked to the packet sender
- o REQ 3: Provided mechanism SHOULD minimise the possibility for cheating
- o REQ 4: Some detection and fallback mechanism SHOULD exist to avoid loss of communication due to the attempted usage of ECN in case an intermediate node clears ECT or drops packets that are ECT marked.

- o REQ 5: Negotiation of ECN SHOULD NOT significantly increase the time taken to negotiate and set-up the RTP session (an extra RTT before the media can flow is unlikely to be acceptable for some use cases).
- o REQ 6: Negotiation of ECN SHOULD NOT cause media clipping at the start of a session.

The following sections describes how these requirements can be meet for RTP over UDP.

### 3.2. Applicability

The use of ECN with RTP over UDP is dependent on negotiation of ECN capability between the sender and receiver(s), and validation of ECN support in all elements of the network path(s) traversed. RTP is used in a heterogeneous range of network environments and topologies, with various different signalling protocols, all of which need to be verified to support ECN before it can be used.

The usage of ECN is further dependent on a capability of the RTP media flow to react to congestion signalled by ECN marked packets. Depending on the application, media codec, and network topology, this adaptation can occur in various forms and at various nodes. As an example, the sender can change the media encoding, or the receiver can change the subscription to a layered encoding, or either reaction can be accomplished by a transcoding middlebox. RFC 5117 identifies seven topologies in which RTP sessions may be configured, and which may affect the ability to use ECN:

**Topo-Point-to-Point:** This is a standard unicast flow. ECN may be used with RTP in this topology in an analogous manner to its use with other unicast transport protocols, with RTCP conveying ECN feedback messages.

**Topo-Multicast:** This is either an any source multicast (ASM) group with potentially several active senders and multicast RTCP feedback, or a source specific multicast (SSM) group with a single sender and unicast RTCP feedback from receivers. RTCP is designed to scale to large group sizes while avoiding feedback implosion (see Section 6.2 of [RFC3550], [RFC4585], and [RFC5760]), and can be used by a sender to determine if all its receivers, and the network paths to those receivers, support ECN (see Section 6.2). It is somewhat more difficult to determine if all network paths from all senders to all receivers support ECN. Accordingly, we allow ECN to be used by an RTP sender using multicast UDP provided the sender has verified that the paths to all its known receivers support ECN, and irrespective of whether the paths from other

senders to their receivers support ECN. Note that group membership may change during the lifetime of a multicast RTP session, potentially introducing new receivers that are not ECN capable. Senders must use the mechanisms described in Section 6.4 to monitor that all receivers continue to support ECN, and they need to fallback to non-ECN use if any senders do not.

**Topo-Translator:** An RTP translator is an RTP-level middlebox that is invisible to the other participants in the RTP session (although it is usually visible in the associated signalling session). There are two types of RTP translator: those do not modify the media stream, and are concerned with transport parameters, for example a multicast to unicast gateway; and those that do modify the media stream, for example transcoding between different media codecs. A single RTP session traverses the translator, and the translator must rewrite RTCP messages passing through it to match the changes it makes to the RTP data packets. A legacy, ECN-unaware, RTP translator is expected to ignore the ECN bits on received packets, and to set the ECN bits to not-ECT when sending packets, so causing ECN negotiation on the path containing the translator to fail (any new RTP translator that does not wish to support ECN may do so similarly). An ECN aware RTP translator may act in one of three ways:

- \* If the translator does not modify the media stream, it should copy the ECN bits unchanged from the incoming to the outgoing datagrams, unless it is overloaded and experiencing congestion, in which case it may mark the outgoing datagrams with an ECN-CE mark. Such a translator passes RTCP feedback unchanged.
- \* If the translator modifies the media stream to combine or split RTP packets, but does not otherwise transcode the media, it must manage the ECN bits in a way analogous to that described in Section 5.3 of [RFC3168]: if an ECN marked packet is split into two, then both the outgoing packets must be ECN marked identically to the original; if several ECN marked packets are combined into one, the outgoing packet must be either ECN-CE marked or dropped if any of the incoming packets are ECN-CE marked. If the outgoing combined packet is not ECN-CE marked, then it MUST be ECT marked if any of the incoming packets were ECT marked. When RTCP ECN feedback packets (Section 5) are received, they must be rewritten to match the modifications made to the media stream (see Section 7.1).
- \* If the translator is a media transcoder, the output RTP media stream may have radically different characteristics than the input RTP media stream. Each side of the translator must then be considered as a separate transport connection, with its own

ECN processing. This requires the translator interpose itself into the ECN negotiation process, effectively splitting the connection into two parts with their own negotiation. Once negotiation has been completed, the translator must generate RTCP ECN feedback back to the source based on its own reception, and must respond to RTCP ECN feedback received from the receiver(s) (see Section 7.2).

It is recognised that ECN and RTCP processing in an RTP translator that modifies the media stream is non-trivial.

**Topo-Mixer:** A mixer is an RTP-level middlebox that aggregates multiple RTP streams, mixing them together to generate a new RTP stream. The mixer is visible to the other participants in the RTP session, and is also usually visible in the associated signalling session. The RTP flows on each side of the mixer are treated independently for ECN purposes, with the mixer generating its own RTCP ECN feedback, and responding to ECN feedback for data it sends. Since connections are treated independently, it would seem reasonable to allow the transport on one side of the mixer to use ECN, while the transport on the other side of the mixer is not ECN capable, if this is desired.

**Topo-Video-switch-MCU:** A video switching MCU receives several RTP flows, but forwards only one of those flows onwards to the other participants at a time. The flow that is forwarded changes during the session, often based on voice activity. Since only a subset of the RTP packets generated by a sender are forwarded to the receivers, a video switching MCU can break ECN negotiation (the success of the ECN negotiation may depend on the voice activity of the participant at the instant the negotiation takes place - shout if you want ECN). It also breaks congestion feedback and response, since RTP packets are dropped by the MCU depending on voice activity rather than network congestion. This topology is widely used in legacy products, but is NOT RECOMMENDED for new implementations and cannot be used with ECN.

**Topo-RTCP-terminating-MCU:** In this scenario, each participant runs an RTP point-to-point session between itself and the MCU. Each of these sessions is treated independently for the purposes of ECN and RTCP feedback, potentially with some using ECN and some not.

**Topo-Asymmetric:** It is theoretically possible to build a middlebox that is a combination of an RTP mixer in one direction and an RTP translator in the other. To quote RFC 5117 "This topology is so problematic and it is so easy to get the RTCP processing wrong, that it is NOT RECOMMENDED to implement this topology."



These topologies may be combined within a single RTP session.

The ECN mechanism defined in this memo is applicable to both sender and receiver controlled congestion algorithms. The mechanism ensures that both senders and receivers will know about ECN-CE markings and any packet losses. Thus the actual decision point for the congestion control is not relevant. This is a great benefit as the rate of an RTP session can be varied in a number of ways, for example a unicast media sender might use TFRC [RFC5348] or some other algorithm, while a multicast session could use a sender based scheme adapting to the lowest common supported rate, or a receiver driven mechanism using layered coding to support more heterogeneous paths.

To ensure timely feedback of CE marked packets, this mechanism requires support for the RTP/AVPF profile [RFC4585] or any of its derivatives, such as RTP/SAVPF [RFC5124]. The standard RTP/AVP profile [RFC3551] does not allow any early or immediate transmission of RTCP feedback, and has a minimal RTCP interval whose default value (5 seconds) is many times the normal RTT between sender and receiver.

The control of which RTP data packets are marked as ECT, and whether ECT(0) or ECT(1) is used, is due to the sender. RTCP packets MUST NOT be ECT marked, whether generated by sender or receivers.

#### 4. Overview of Use of ECN with RTP/UDP/IP

The solution for using ECN with RTP over UDP/IP consists of four different pieces that together make the solution work:

1. Negotiation of the capability to use ECN with RTP/UDP/IP
2. Initiation and initial verification of ECN capable transport
3. Ongoing use of ECN within an RTP session
4. Handling of dynamic groups through failure detection, verification and fallback

The solution includes a new SDP attribute (Section 6.1.1), the definition of new extensions to RTCP (Section 5) and STUN (Section 6.2.2).

Before an RTP session can be created, a signalling protocol is used to discover the other participants and negotiate session parameters (see Section 6.1). One of the parameters that can be negotiated is the capability of a participant to support ECN functionality, or otherwise. Note that all participants having the capability of

supporting ECN does not necessarily imply that ECN is usable in an RTP session, since there may be middleboxes on the path between the participants which don't pass ECN-marked packets (for example, a firewall that blocks traffic with the ECN bits set). This document defines the information that needs to be negotiated, and provides a mapping to SDP for use in both declarative and offer/answer contexts.

When a sender joins a session for which all participants claim ECN capability, it must verify if that capability is usable. There are three ways in which this verification may be done (Section 6.2):

- o The sender may generate a (small) subset of its RTP data packets with the ECN field set to ECT(0) or ECT(1). Each receiver will then send an RTCP feedback packet indicating the reception of the ECT marked RTP packets. Upon reception of this feedback from each receiver it knows of, the sender can consider ECN functional for its traffic. Each sender does this verification independently of each other. If a new receiver joins an existing session it will reveal whether or not it supports ECN when it sends its first RTCP report to each source. If the RTCP report includes ECN information, verification will have succeeded and sources can continue to send ECT packets. If not, verification fails and each sender MUST stop using ECN.
- o Alternatively, ECN support can be verified during an initial end-to-end STUN exchange (for example, as part of ICE connection establishment). After having verified connectivity without ECN capability an extra STUN exchange, this time with the ECN field set to ECT(0) or ECT(1), is performed. If successful the path's capability to convey ECN marked packets is verified. A new STUN attribute is defined to convey feedback that the ECT marked STUN request was received (see Section 9.4), along with an ICE signalling option (Section 9.5).
- o Thirdly, the sender may make a leap of faith that ECN will work. This is only recommended for applications that know they are running in controlled environments where ECN functionality has been verified through other means. In this mode it is assumed that ECN works, and the system reacts to failure indicators if the assumption proved wrong. The use of this method relies on a high confidence that ECN operation will be successful, or an application where failure is not serious. The impact on the network and other users must be considered when making a leap of faith, so there are limitations on when this method is allowed.

The first mechanism, using RTP with RTCP feedback, has the advantage of working for all RTP sessions, but the disadvantages of potential clipping if ECN marked RTP packets are discarded by middleboxes, and

slow verification of ECN support. The STUN-based mechanism is faster to verify ECN support, but only works in those scenarios supported by end-to-end STUN, such as within an ICE exchange. The third one, leap-of-faith, has the advantage of avoiding additional tests or complexities and enabling ECN usage from the first media packet. The downside is that if the end-to-end path contains middleboxes that do not pass ECN, the impact on the application can be severe: in the worst case, all media could be lost if a middlebox that discards ECN marked packets is present. A less severe effect, but still requiring reaction, is the presence of a middlebox that re-marks ECT marked packets to non-ECT, possibly marking packets with a CE mark as non-ECT. This can force the network into heavy congestion due to non-responsiveness, and seriously impact media quality.

Once ECN support has been verified (or assumed) to work for all receivers, a sender marks all its RTP packets as ECT packets, while receivers rapidly feedback any CE marks to the sender using RTCP in RTP/AVPF immediate or early feedback mode. An RTCP feedback report is sent as soon as possible by the transmission rules for feedback that are in place. This feedback report indicates the receipt of new CE marks since the last ECN feedback packet, and also counts the total number of CE marked packets through a cumulative sum. This is the mechanism to provide the fastest possible feedback to senders about CE marks. On receipt of a CE marked packet, the system must react to congestion as-if packet loss has been reported. Section 6.3 describes the ongoing use of ECN with an RTP session.

This rapid feedback is not optimised for reliability, therefore an additional procedure, the RTCP ECN summary reports, is used to ensure more reliable, but less timely, reporting of the ECN information. The ECN summary report contains the same information as the ECN feedback format, only packed differently for better efficiency with large reports. It is sent in a compound RTCP packet, along with regular RTCP reception reports. By using cumulative counters for seen CE, ECT, not-ECT and packet loss the sender can determine what events have happened since the last report, independently of any RTCP packets having been lost.

RTCP traffic MUST NOT be ECT marked for the following reason. ECT marked traffic may be dropped if the path is not ECN compliant. As RTCP is used to provide feedback about what has been transmitted and what ECN markings that are received, it is important that these are received in cases when ECT marked traffic is not getting through.

There are numerous reasons why the path the RTP packets take from the sender to the receiver may change, e.g., mobility, link failure followed by re-routing around it. Such an event may result in the packet being sent through a node that is ECN non-compliant, thus re-

marking or dropping packets with ECT set. To prevent this from impacting the application for longer than necessary, the operation of ECN is constantly monitored by all senders. Both the RTCP ECN summary reports and the ECN feedback packets allow the sender to compare the number of ECT(0), ECT(1), and non-ECT marked packets received with the number that were sent, while also reporting CE marked and lost packets. If these numbers do not agree, it can be inferred that the path does not reliably pass ECN-marked packets (Section 6.4.2 discusses how to interpret the different cases). A sender detecting a possible ECN non-compliance issue should then stop sending ECT marked packets to determine if that allows the packets to be correctly delivered. If the issues can be connected to ECN, then ECN usage is suspended and possibly also re-negotiated.

## 5. RTCP Extensions for ECN feedback

This documents defines two different RTCP extensions: one RTP/AVPF [RFC4585] transport layer feedback format for urgent ECN information, and one RTCP XR [RFC3611] ECN summary report block type for regular reporting of the ECN marking information. The full definition of these extensions usage as part of the complete solution is laid out in Section 6.

### 5.1. RTP/AVPF Transport Layer ECN Feedback packet

This RTP/AVPF transport layer feedback format is intended for usage in AVPF early or immediate feedback modes when information needs to urgently reach the sender. Thus its main use is to report on reception of an ECN-CE marked RTP packet so that the sender may perform congestion control, or to speed up the initiation procedures by rapidly reporting that the path can support ECN-marked traffic. The feedback format is also defined with reduced size RTCP [RFC5506] in mind, where RTCP feedback packets may be sent without accompanying Sender or Receiver Reports that would contain the Extended Highest Sequence number and the accumulated number of packet losses. Both are important for ECN to verify functionality and keep track of when CE marking does occur.

The RTP/AVPF transport layer feedback packet starts with the common header defined by the RTP/AVPF profile [RFC4585] which is reproduced here for the reader's information:

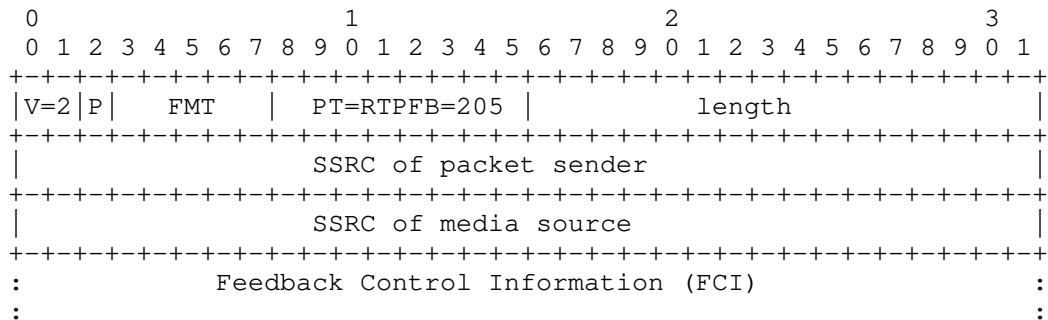


Figure 1: RTP/AVPF Common Packet Format for Feedback Messages

From Figure 1 it can be determined the identity of the feedback provider and for which RTP packet sender it applies. Below is the feedback information format defined that is inserted as FCI for this particular feedback messages that is identified with an FMT value = 6.

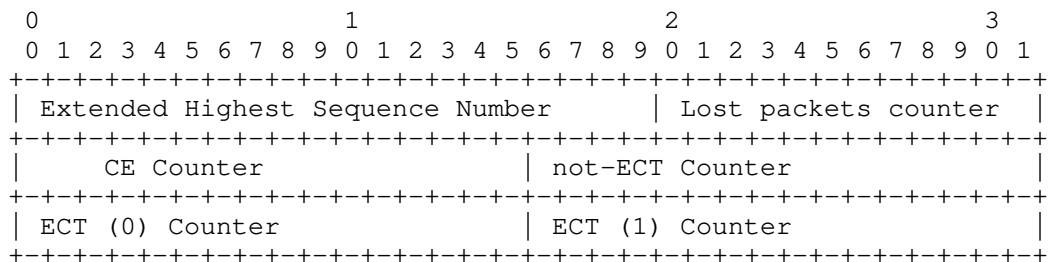


Figure 2: ECN Feedback Format

The FCI information for the ECN Feedback format (Figure 2) are the following:

**Extended Highest Sequence Number:** The least significant 20-bit from an Extended highest sequence number received value as defined by [RFC3550]. Used to indicate for which packet this report is valid up to.

**Lost Packets Counter:** The cumulative number of RTP packets that the receiver expected to receive from this SSRC, minus the number of packets it actually received. This is the same as the cumulative number of packets lost defined in Section 6.4.1 of [RFC3550] except represented in 12-bit signed format, compared to 24-bit in RTCP SR or RR packets. As with the equivalent value in RTCP SR or RR packets, note that packets that arrive late are not counted as lost, and the loss may be negative if there are duplicates.

**CE Counter:** The cumulative number of RTP packets received from this SSRC since the receiver joined the RTP session that were ECN-CE marked. The receiver should keep track of this value using a local representation that is longer than 16-bits, and only include the 16-bits with least significance. In other words, the field will wrap to 0 if more than 65535 packets has been received.

**ECT(0) Counter:** The cumulative number of RTP packets received from this SSRC since the receiver joined the RTP session that had an ECN field value of ECT(0). The receiver should keep track of this value using a local representation that is longer than 16-bits, and only include the 16-bits with least significance. In other words, the field will wrap if more than 65535 packets have been received.

**ECT(1) Counter:** The cumulative number of RTP packets received from this SSRC since the receiver joined the RTP session that had an ECN field value of ECT(1). The receiver should keep track of this value using a local representation that is longer than 16-bits, and only include the 16-bits with least significance. In other words, the field will wrap if more than 65535 packets have been received.

**not-ECT Counter:** The cumulative number of RTP packets received from this SSRC since the receiver joined the RTP session that had an ECN field value of not-ECT. The receiver should keep track of this value using a local representation that is longer than 16-bits, and only include the 16-bits with least significance. In other words, the field will wrap if more than 65535 packets have been received.

Each FCI block reports on a single source (SSRC). Multiple sources can be reported by including multiple RTCP feedback messages in a compound RTCP packet. The AVPF common header indicates both the sender of the feedback message and on which stream it relates to.

The Counters SHALL be initiated to 0 for a new receiver. This to enable detection of CE or Packet loss already on the initial report from a specific participant.

The Extended Highest sequence number and packet loss fields are both truncated in comparison to the RTCP SR or RR versions. This is to save bits as the representation is redundant unless reduced size RTCP is used in such a way that only feedback packets are transmitted, with no SR or RR in the compound RTCP packet. Due to that regular RTCP reporting will include the longer versions of the fields the wrapping issue will be less unless the packet rate of the application is so high that the fields will wrap within a regular RTCP reporting

interval. In those case the feedback packet need to be sent in a compound packet together with the SR or RR packet.

There is an issue with packet duplication in relation to the packet loss counter. If one avoids holding state for which sequence number has been received then the way one can count loss is to count the number of received packets and compare that to the number of packets expected. As a result a packet duplication can hide a packet loss. If a receiver is tracking the sequence numbers actually received and suppresses duplicates it provides for a more reliable packet loss indication. Reordering may also result in that packet loss is reported in one report and then removed in the next.

The CE counter is actually more robust for packet duplication. Adding each received CE marked packet to the counter is not an issue. If one of the clones was CE marked that is still a indication of congestion. Packet duplication has potential impact on the ECN verification. Thus the sum of packets reported may be higher than the number sent. However, most detections are still applicable.

## 5.2. RTCP XR Report block for ECN summary information

This report block combined with RTCP SR or RR report blocks carries the same information as the ECN Feedback Packet and shall be based on the same underlying information. However, there is a difference in semantics between the feedback format and this XR version. Where the feedback format is intended to report on a CE mark as soon as possible, this extended report is for the regular RTCP report and continuous verification of the ECN functionality end-to-end.

The ECN Summary report block consists of one report block header:

```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|          BT          |  Reserved  |          Block Length          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

and then followed of one or more of the following report data blocks:

```

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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| SSRC of Media Sender |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| CE Counter          | not-ECT Counter          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| ECT (0) Counter     | ECT (1) Counter         |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

BT: Block Type identifying the ECN summary report block. Value is 13.

Reserved: All bits SHALL be set to 0 on transmission and ignored on reception.

Block Length: The length of the report block. Used to indicate the number of report data blocks present in the ECN summary report. This length will be  $3*n$ , where  $n$  is the number of ECN summary report blocks, since blocks are a fixed size.

SSRC of Media Sender: The SSRC identifying the media sender this report is for.

CE Counter: as in Section 5.1.

ECT(0) Counter: as in Section 5.1.

ECT(1) Counter: as in Section 5.1.

not-ECT Counter: as in Section 5.1.

The Extended Highest Sequence number and the packet loss counter for each SSRC is not present in RTCP XR report, in contrast to the feedback version. The reason is that this summary report will always be sent in a RTCP compound packet where the Extended Highest Sequence number and the accumulated number of packet losses are present in the RTCP Sender Report or Receiver Report packet's report block.

## 6. Use of ECN with RTP/UDP/IP

In the detailed specification of the behaviour below, the different functions in the general case will first be discussed. In case special considerations are needed for middleboxes, multicast usage etc, those will be specially discussed in related subsections.

### 6.1. Negotiation of ECN Capability

The first stage of ECN negotiation for RTP-over-UDP is to signal the capability to use ECN. This includes negotiating if ECN is to be used symmetrically and the method for initial ECT verification. This memo defines the mappings of this information onto SDP for both declarative and offer/answer usage. There is one SDP extension to indicate if ECN support should be used, and the method for initiation. In addition an ICE parameter is defined to indicate that ECN initiation using STUN is supported as part of an ICE exchange.



An RTP system that supports ECN and uses SDP in the signalling MUST implement the SDP extension to signal ECN capability as described in Section 6.1.1. It MAY also implement alternative ECN capability negotiation schemes, such as the ICE extension described in Section 6.1.2.

#### 6.1.1. Signalling ECN Capability using SDP

One new SDP attribute, "a=ecn-capable-rtp", is defined. This is a media level attribute, which MUST NOT be used at the session level. It is not subject to the character set chosen. The aim of this signalling is to indicate the capability of the sender and receivers to support ECN, and to negotiate the method of ECN initiation to be used in the session. The attribute takes a list of initiation methods, ordered in decreasing preference. The defined values for the initiation method are:

rtp: Using RTP and RTCP as defined in Section 6.2.1.

ice: Using STUN within ICE as defined in Section 6.2.2.

leap: Using the leap of faith method as defined in Section 6.2.3.

In addition, a number of OPTIONAL parameters may be included in the "a=ecn-capable-rtp" attribute as follows:

mode: This parameter signals the endpoint's capability to set and read ECN marks in UDP packets. An examination of various operating systems has shown that end-system support for ECN marking of UDP packets may be symmetric or asymmetric. By this we mean that some systems may allow end points to set the ECN bits in an outgoing UDP packet but not read them, while others may allow applications to read the ECN bits but not set them. This either/or case may produce an asymmetric support for ECN and thus should be conveyed in the SDP signalling. The "mode=setread" state is the ideal condition where an endpoint can both set and read ECN bits in UDP packets. The "mode=setonly" state indicates that an endpoint can set the ECT bit, but cannot read the ECN bits from received UDP packets to determine if upstream congestion occurred. The "mode=readonly" state indicates that the endpoint can read the ECN bits to determine if downstream congestion has occurred, but it cannot set the ECT bits in outgoing UDP packets. When the "mode=" parameter is omitted it is assumed that the node has "setread" capabilities. This option can provide for an early indication that ECN cannot be used in a session. This would be case when both the offerer and answerer set the "mode=" parameter to "setonly" or "readonly", or when an RTP sender entity considers offering "readonly".

ect: This parameter makes it possible to express the preferred ECT marking. This is either "random", "0", or "1", with "0" being implied if not specified. The "ect" parameter describes a receiver preference, and is useful in the case where the receiver knows it is behind a link using IP header compression, the efficiency of which would be seriously disrupted if it were to receive packets with randomly chosen ECT marks. It is RECOMMENDED that ECT(0) marking be used.

The ABNF [RFC5234] grammar for the "a=ecn-capable-rtp" attribute is as follows:

```

ecn-attribute  = "a=ecn-capable-rtp:" SP init-list SP parm-list
init-list     = init-value *("," init-value)
init-value    = "rtp" / "ice" / "leap" / init-ext
init-ext      = token
parm-list     = parm-value *("; " SP parm-value)
parm-value    = mode / ect / parm-ext
mode          = "mode=" ("setonly" / "setread" / "readonly")
ect           = "ect=" ("0" / "1")
parm-ext      = parm-name "=" parm-value-ext
parm-name     = token
parm-value-ext = token / quoted-string
quoted-string = DQUOTE *qdttext DQUOTE
qdttext       = %x20-21 / %x23-7E / %x80-FF
               ; any 8-bit ascii except "<">

; external references:
; token: from RFC 4566
; SP and DQUOTE from RFC 5234

```

When SDP is used with the offer/answer model [RFC3264], the party generating the SDP offer MUST insert an "a=ecn-capable-rtp" attribute into the media section of the SDP offer of each RTP flow for which it wishes to use ECN. The attribute includes one or more ECN initiation methods in a comma separated list in decreasing order of preference, with some number of optional parameters following. The answering party compares the list of initiation methods in the offer with those it supports in order of preference. If there is a match, and if the receiver wishes to attempt to use ECN in the session, it includes an "a=ecn-capable-rtp" attribute containing its single preferred choice of initiation method in the media sections of the answer. If there is no matching initiation method capability, or if the receiver does not wish to attempt to use ECN in the session, it does not include an "a=ecn-capable-rtp" attribute in its answer. If the attribute is removed in the answer then ECN MUST NOT be used in any direction for that media flow. The answer may also include optional parameters, as discussed below.

If the "mode=setonly" parameter is present in the "a=ecn-capable-rtp" attribute of the offer and the answering party is also "mode=setonly", then there is no common ECN capability, and the answer MUST NOT include the "a=ecn-capable-rtp" attribute. Otherwise, if the offer is "mode=setonly" then ECN may only be initiated in the direction from the offering party to the answering party.

If the "mode=readonly" parameter is present in the "a=ecn-capable-rtp" attribute of the offer and the answering party is "mode=readonly", then there is no common ECN capability, and the answer MUST NOT include the "a=ecn-capable-rtp" attribute. Otherwise, if the offer is "mode=readonly" then ECN may only be initiated in the direction from the answering party to the offering party.

If the "mode=setread" parameter is present in the "a=ecn-capable-rtp" attribute of the offer and the answering party is "setonly", then ECN may only be initiated in the direction from the answering party to the offering party. If the offering party is "mode=setread" but the answering party is "mode=readonly", then ECN may only be initiated in the direction from the offering party to the answering party. If both offer and answer are "mode=setread", then ECN may be initiated in both directions. Note that "mode=setread" is implied by the absence of a "mode=" parameter in the offer or the answer.

The "ect=" parameter in the "a=ecn-capable-rtp" attribute is set independently in the offer and the answer. Its value in the offer indicates a preference for the behaviour of the answering party, and its value in the answer indicates a preference for the behaviour of the offering party. It will be the senders choice if to honor the receivers preference or not.

When SDP is used in a declarative manner, for example in a multicast session using the Session Announcement Protocol (SAP, [RFC2974]), negotiation of session description parameters is not possible. The "a=ecn-capable-rtp" attribute MAY be added to the session description to indicate that the sender will use ECN in the RTP session. The attribute MUST include a single method of initiation. Participants MUST NOT join such a session unless they have the capability to understand ECN-marked UDP packets, implement the method of initiation, and can generate RTCP ECN feedback (note that having the capability to use ECN doesn't necessarily imply that the underlying network path between sender and receiver supports ECN). The mode parameter MAY be included also in declarative usage, to indicate which capability is required by the consumer of the SDP. So for example in a SSM session the participants configured with a particular SDP will all be in a media receive only mode, thus

mode=readonly will work as the capability of reporting on the ECN markings in the received is what is required.

The "a=ecn-capable-rtp" attribute MAY be used with RTP media sessions using UDP/IP transport. It MUST NOT be used for RTP sessions using TCP, SCTP, or DCCP transport, or for non-RTP sessions.

As described in Section 6.3.3, RTP sessions using ECN require rapid RTCP ECN feedback, in order that the sender can react to ECN-CE marked packets. Thus, the use of the Extended RTP Profile for RTCP-Based Feedback (RTP/AVPF) [RFC4585] or other profile that inherits AVPF's signalling rules, MUST be signalled.

#### 6.1.2. ICE Parameter to Signal ECN Capability

One new ICE [RFC5245] option, "rtp+ecn", is defined. This is used with the SDP session level "a=ice-options" attribute in an SDP offer to indicate that the initiator of the ICE exchange has the capability to support ECN for RTP-over-UDP flows (via "a=ice-options: rtp+ecn"). The answering party includes this same attribute at the session level in the SDP answer if it also has the capability, and removes the attribute if it does not wish to use ECN, or doesn't have the capability to use ECN. If this initiation method (Section 6.2.2) actually is going to be used, it is explicitly negotiated using the "a=ecn-capable-rtp" attribute.

Note: This signalling mechanism is not strictly needed as long as the STUN ECN testing capability is used within the context of this document. It may however be useful if the ECN verification capability is used in additional contexts.

#### 6.2. Initiation of ECN Use in an RTP Session

Once the sender and the receiver(s) have agreed that they have the capability to use ECN within a session, they may attempt to initiate ECN use.

At the start of the RTP session, when the first packets with ECT are sent, it is important to verify that IP packets with ECN field values of ECT or ECN-CE will reach their destination(s). There is some risk that the use of ECN will result in either reset of the ECN field, or loss of all packets with ECT or ECN-CE markings. If the path between the sender and the receivers exhibits either of these behaviours one needs to stop using ECN immediately to protect both the network and the application.

The RTP senders and receivers SHALL NOT ECT mark their RTCP traffic at any time. This is to ensure that packet loss due to ECN marking

will not effect the RTCP traffic and the necessary feedback information it carries.

An RTP system that supports ECN MUST implement the initiation of ECN using in-band RTP and RTCP described in Section 6.2.1. It MAY also implement other mechanisms to initiate ECN support, for example the STUN-based mechanism described in Section 6.2.2 or use the leap of faith option if the session supports the limitations provided in Section 6.2.3. If support for both in-band and out-of-band mechanisms is signalled, the sender should try ECN negotiation using STUN with ICE first, and if it fails, fallback to negotiation using RTP and RTCP ECN feedback.

No matter how ECN usage is initiated, the sender MUST continually monitor the ability of the network, and all its receivers, to support ECN, following the mechanisms described in Section 6.4. This is necessary because path changes or changes in the receiver population may invalidate the ability of the system to use ECN.

#### 6.2.1. Detection of ECT using RTP and RTCP

The ECN initiation phase using RTP and RTCP to detect if the network path supports ECN comprises three stages. Firstly, the RTP sender generates some small fraction of its traffic with ECT marks to act a probe for ECN support. Then, on receipt of these ECT-marked packets, the receivers send RTCP ECN feedback packets and RTCP ECN summary reports to inform the sender that their path supports ECN. Finally, the RTP sender makes the decision to use ECN or not, based on whether the paths to all RTP receivers have been verified to support ECN.

**Generating ECN Probe Packets:** During the ECN initiation phase, an RTP sender SHALL mark a small fraction of its RTP traffic as ECT, while leaving the remainder of the packets unmarked. The main reason for only marking some packets is to maintain usable media delivery during the ECN initiation phase in those cases where ECN is not supported by the network path. A secondary reason to send some not-ECT packets are to ensure that the receivers will send RTCP reports on this sender, even if all ECT marked packets are lost in transit. The not-ECT packets also provide a base-line to compare performance parameters against. A fourth reason for only probing with a small number of packets is to reduce the risk that significant numbers of congestion markings might be lost if ECT is cleared to Not-ECT by an ECN-Reverting Meddlebox. Then any resulting lack of congestion response is likely to have little damaging affect on others. An RTP sender is RECOMMENDED to send a minimum of two packets with ECT markings per RTCP reporting interval, one with ECT(0) and one with ECT(1), and will continue to send some ECT marked traffic as long as the ECN initiation

phase continues. The sender SHOULD NOT mark all RTP packets as ECT during the ECN initiation phase.

This memo does not mandate which RTP packets are marked with ECT during the ECN initiation phase. An implementation should insert ECT marks in RTP packets in a way that minimises the impact on media quality if those packets are lost. The choice of packets to mark is clearly very media dependent, but the usage of RTP NO-OP payloads [I-D.ietf-avt-rtp-no-op], if supported, would be an appropriate choice. For audio formats, it would make sense for the sender to mark comfort noise packets or similar. For video formats, packets containing P- or B-frames, rather than I-frames, would be an appropriate choice. No matter which RTP packets are marked, those packets MUST NOT be duplicated in transmission, since their RTP sequence number is used to identify packets that are received with ECN markings.

**Generating RTCP ECN Feedback:** If ECN capability has been negotiated in an RTP session, the receivers in the session MUST listen for ECT or ECN-CE marked RTP packets, and generate RTCP ECN feedback packets (Section 5.1) to mark their receipt. An immediate or early (depending on the RTP/AVPF mode) ECN feedback packet SHOULD be generated on receipt of the first ECT or ECN-CE marked packet from a sender that has not previously sent any ECT traffic. Each regular RTCP report MUST also contain an ECN summary report (Section 5.2). Reception of subsequent ECN-CE marked packets SHOULD result in additional early or immediate ECN feedback packets being sent.

**Determination of ECN Support:** RTP is a group communication protocol, where members can join and leave the group at any time. This complicates the ECN initiation phase, since the sender must wait until it believes the group membership has stabilised before it can determine if the paths to all receivers support ECN (group membership changes after the ECN initiation phase has completed are discussed in Section 6.3).

An RTP sender shall consider the group membership to be stable after it has been in the session and sending ECT-marked probe packets for at least three RTCP reporting intervals (i.e., after sending its third regularly scheduled RTCP packet), and when a complete RTCP reporting interval has passed without changes to the group membership. ECN initiation is considered successful when the group membership is stable, and all known participants have sent one or more RTCP ECN feedback packets indicating correct receipt of the ECT-marked RTP packets generated by the sender.

As an optimisation, if an RTP sender is initiating ECN usage towards a unicast address, then it MAY treat the ECN initiation as provisionally successful if it receives a single RTCP ECN feedback report indicating successful receipt of the ECT-marked packets, with no negative indications, from a single RTP receiver. After declaring provisional success, the sender MAY generate ECT-marked packets as described in Section 6.3, provided it continues to monitor the RTCP reports for a period of three RTCP reporting intervals from the time the ECN initiation started, to check if there is any other participants in the session. If other participants are detected, the sender MUST fallback to only ECT-marking a small fraction of its RTP packets, while it determines if ECN can be supported following the full procedure described above.

Note: One use case that requires further consideration is a unicast connection with several SSRCs multiplexed onto the same flow (e.g., an SVC video using SSRC multiplexing for the layers). It is desirable to be able to rapidly negotiate ECN support for such a session, but the optimisation above fails since the multiple SSRCs make it appear that this is a group communication scenario. It's not sufficient to check that all SSRCs map to a common RTCP CNAME to check if they're actually located on the same device, because there are implementations that use the same CNAME for different parts of a distributed implementation.

ECN initiation is considered to have failed at the instant when any RTP session participant sends an RTCP packet that doesn't contain an RTCP ECN feedback report or ECN summary report, but has an RTCP RR with an extended RTP sequence number field that indicates that it should have received multiple (>3) ECT marked RTP packets. This can be due to failure to support the ECN feedback format by the receiver or some middlebox, or the loss of all ECT marked packets. Both indicate a lack of ECN support.

If the ECN negotiation succeeds, this indicates that the path can pass some ECN-marked traffic, and that the receivers support ECN feedback. This does not necessarily imply that the path can robustly convey ECN feedback; Section 6.3 describes the ongoing monitoring that must be performed to ensure the path continues to robustly support ECN.

When a sender or receiver detects ECN failures on paths they should log these to enable follow up and statistics gathering regarding broken paths. The logging mechanism used is implementation dependent.

## 6.2.2. Detection of ECT using STUN with ICE

This section describes an OPTIONAL method that can be used to avoid media impact and also ensure an ECN capable path prior to media transmission. This method is considered in the context where the session participants are using ICE [RFC5245] to find working connectivity. We need to use ICE rather than STUN only, as the verification needs to happen from the media sender to the address and port on which the receiver is listening.

To minimise the impact of set-up delay, and to prioritise the fact that one has a working connectivity rather than necessarily finding the best ECN capable network path, this procedure is applied after having performed a successful connectivity check for a candidate, which is nominated for usage. At that point, and provided the chosen candidate is not a relayed address, an additional connectivity check is performed, sending the "ECT Check" attribute in a STUN packet that is ECT marked. On reception of the packet, a STUN server supporting this extension will note the received ECN field value, and send a STUN/UDP/IP packet in reply, with the ECN field set to not-ECT, and including an ECN check attribute. A STUN server that doesn't understand the extension or are incapable of reading the ECN values on incoming STUN packets SHALL follow the STUN specifications rule for unknown comprehension-required attributes, i.e. send a 420 (Unknown Attribute) response back.

The STUN ECN check attribute contains one field and a flag. The flag indicates if the echo field contains a valid value or not. The field is the ECN echo field, and when valid contains the two ECN bits from the packet it echoes back. The ECN check attribute is a comprehension optional attribute.

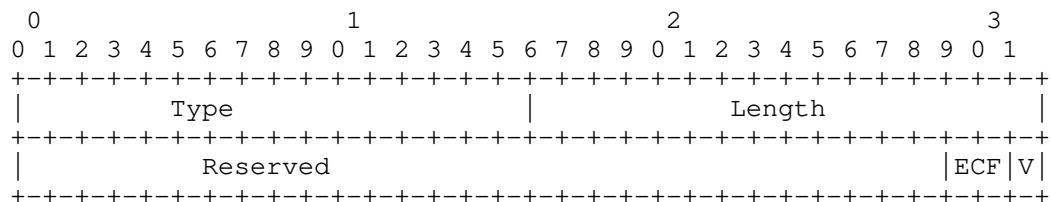


Figure 3: ECN Check STUN Attribute

V: Valid (1 bit) ECN Echo value field is valid when set to 1, and invalid when set 0.



ECF: ECN Echo value field (2 bits) contains the ECN field value of the STUN packet it echoes back when field is valid. If invalid the content is arbitrary.

Reserved: Reserved bits (29 bits) SHALL be set to 0 on transmission, and SHALL be ignored on reception.

This attribute MAY be included in any STUN request to request the ECN field to be echoed back. In STUN requests the V bit SHALL be set to 0. A STUN server receiving a request with the ECN Check attribute which understand it SHALL read the ECN field value of the IP/UDP packet the request was received in. Upon forming the response the server SHALL include the ECN Check attribute setting the V bit to valid and include the read value of the ECN field into the ECF field. If the STUN responder was unable to ascertain due to temporary errors the ECN value of the STUN request, it SHALL set the V bit in the response to 0. The STUN client may retry immediately.

#### 6.2.3. Leap of Faith ECT initiation method

This method for initiating ECN usage is a leap of faith that assumes that ECN will work on the used path(s). The method is to go directly to "ongoing use of ECN" as defined in Section 6.3. Thus all RTP packets MAY be marked as ECT and the failure detection MUST be used to detect any case when the assumption that the path was ECT capable is wrong. This method is only recommended for controlled environments where the whole path(s) between sender and receiver(s) has been built and verified to be ECT.

If the sender marks all packets as ECT while transmitting on a path that contains an ECN-blocking middlebox, then receivers downstream of that middlebox will not receive any RTP data packets from the sender, and hence will not consider it to be an active RTP SSRC. The sender can detect this and revert to sending packets without ECT marks, since RTCP SR/RR packets from such receivers will either not include a report for sender's SSRC, or will report that no packets have been received, but this takes at least one RTCP reporting interval. It should be noted that a receiver might generate its first RTCP packet immediately on joining a unicast session, or very shortly after joining a RTP/AVPF session, before it has had chance to receive any data packets. A sender that receives RTCP SR/RR packet indicating lack of reception by a receiver SHOULD therefore wait for a second RTCP report from that receiver to be sure that the lack of reception is due to ECT-marking. Since this recovery process can take several tens of seconds, during which time the RTP session is unusable for media, it is NOT RECOMMENDED that the leap-of-faith ECT initiation method be used in environments where ECN-blocking middleboxes are likely to be present.

### 6.3. Ongoing Use of ECN Within an RTP Session

Once ECN usage has been successfully initiated for an RTP sender, that sender begins sending all RTP data packets as ECT-marked, and its receivers continue sending ECN feedback information via RTCP packets. This section describes procedures for sending ECT-marked data, providing ECN feedback information via RTCP, responding to ECN feedback information, and detecting failures and misbehaving receivers.

#### 6.3.1. Transmission of ECT-marked RTP Packets

After a sender has successfully initiated ECN usage, it SHOULD mark all the RTP data packets it sends as ECT. The sender SHOULD mark packets as ECT(0) unless the receiver expresses a preference for ECT(1) using the "ect" parameter in the "a=ecn-capable-rtp" attribute.

The sender SHALL NOT include ECT marks on outgoing RTCP packets, and SHOULD NOT include ECT marks on any other outgoing control messages (e.g. STUN [RFC5389] packets, DTLS [RFC4347] handshake packets, or ZRTP [I-D.zimmermann-avt-zrtcp] control packets) that are multiplexed on the same UDP port. For control packets there might be exceptions, like the STUN based ECN check defined in Section 6.2.2.

#### 6.3.2. Reporting ECN Feedback via RTCP

An RTP receiver that receives a packet with an ECN-CE mark, or that detects a packet loss, MUST schedule the transmission of an RTCP ECN feedback packet as soon as possible (subject to the constraints of [RFC4585] and [RFC3550]) to report this back to the sender. There should be no difference in behavior if ECN-CE marks or packet drops are detected. The feedback RTCP packet sent SHALL consist of at least one ECN feedback packet (Section 5) reporting on the packets received since the last ECN feedback packet, and SHOULD contain an RTCP SR or RR packet. The RTP/AVPF profile in early or immediate feedback mode SHOULD be used where possible, to reduce the interval before feedback can be sent. To reduce the size of the feedback message, reduced size RTCP [RFC5506] MAY be used if supported by the end-points. Both RTP/AVPF and reduced size RTCP MUST be negotiated in the session set-up signalling before they can be used.

Every time a regular compound RTCP packet is to be transmitted, an ECN-capable RTP receiver MUST include an RTCP XR ECN summary report as described in Section 5.2 as part of the compound packet.

The multicast feedback implosion problem, that occurs when many receivers simultaneously send feedback to a single sender, must also

be considered. The RTP/AVPF transmission rules will limit the amount of feedback that can be sent, avoiding the implosion problem but also delaying feedback by varying degrees from nothing up to a full RTCP reporting interval. As a result, the full extent of a congestion situation may take some time to reach the sender, although some feedback should arrive in a reasonably timely manner, allowing the sender to react on a single or a few reports.

A possible future optimisation might be to define some form of feedback suppression mechanism to reduce the RTCP reporting overhead for group communication using ECN.

In case a receiver driven congestion control algorithm is to be used and has been agreed upon through signalling, the algorithm MAY specify that the immediate scheduling (and later transmission) of ECN-CE feedback of any received ECN-CE mark is not required and shall not be done (since it is not necessary for congestion control purposes in such cases). In that case ECN feedback is only sent using regular RTCP reports for verification purpose and in response to the initiation process ("rtp") of any new media senders as specified in Section 6.2.1.

#### 6.3.3. Response to Congestion Notifications

When RTP packets are received with ECN-CE marks, the sender and/or receivers MUST react with congestion control as-if those packets had been lost. Depending on the media format, type of session, and RTP topology used, there are several different types of congestion control that can be used.

**Sender-Driven Congestion Control:** The sender may be responsible for adapting the transmitted bit-rate in response to RTCP ECN feedback. When the sender receives the ECN feedback data it feeds this information into its congestion control or bit-rate adaptation mechanism so that it can react on it as if it was packet losses that was reported. The congestion control algorithm to be used is not specified here, although TFRC [RFC5348] is one example that might be used.

**Receiver-Driven Congestion Control:** If a receiver driven congestion control mechanism is used, the receiver can react to the ECN-CE marks without contacting the sender. This may allow faster response than sender-driven congestion control in some circumstances. Receiver-driven congestion control is usually implemented by providing the content in a layered way, with each layer providing improved media quality but also increased bandwidth usage. The receiver locally monitors the ECN-CE marks on received packet to check if it experiences congestion at the

current number of layers. If congestion is experienced, the receiver drops one layer, so reducing the resource consumption on the path towards itself. For example, if a layered media encoding scheme such as H.264 SVC is used, the receiver may change its layer subscription, and so reduce the bit rate it receives. The receiver MUST still send RTCP ECN feedback to the sender, even if it can adapt without contact with the sender, so that the sender can determine if ECN is supported on the network path. The timeliness of RTCP feedback is less of a concern with receiver driven congestion control, and regular RTCP reporting of ECN feedback is sufficient (without using RTP/AVPF immediate or early feedback).

Responding to congestion indication in the case of multicast traffic is a more complex problem than for unicast traffic. The fundamental problem is diverse paths, i.e. when different receivers don't see the same path, and thus have different bottlenecks, so the receivers may get ECN-CE marked packets due to congestion at different points in the network. This is problematic for sender driven congestion control, since when receivers are heterogeneous in regards to capacity the sender is limited to transmitting at the rate the slowest receiver can support. This often becomes a significant limitation as group size grows. Also, as group size increases the frequency of reports from each receiver decreases, which further reduces the responsiveness of the mechanism. Receiver-driven congestion control has the advantage that each receiver can choose the appropriate rate for its network path, rather than all having to settle for the lowest common rate.

We note that ECN support is not a silver bullet to improving performance. The use of ECN gives the chance to respond to congestion before packets are dropped in the network, improving the user experience by allowing the RTP application to control how the quality is reduced. An application which ignores ECN congestion experienced feedback is not immune to congestion: the network will eventually begin to discard packets if traffic doesn't respond. It is in the best interest of an application to respond to ECN congestion feedback promptly, to avoid packet loss.

#### 6.4. Detecting Failures

Senders and receivers can deliberately ignore ECN-CE and thus get a benefit over behaving flows (cheating). Nonce [RFC3540] is an addition to TCP that solves this issue as long as the sender acts on behalf of the network. The assumption about the senders acting on the behalf of the network may be reduced due to the nature of peer-to-peer use of RTP. Still a significant portion of RTP senders are infrastructure devices (for example, streaming media servers) that do

have an interest in protecting both service quality and the network. Even though there may be cases where nonce can be applicable also for RTP, it is not included in this specification. It is however worth mention that, as real-time media is commonly sensitive to increased delay and packet loss, it will be in both media sender and receivers interest to minimise the number and duration of any congestion events as they will affect media quality.

RTP sessions can also suffer from path changes resulting in a non-ECN compliant node becoming part of the path. That node may perform either of two actions that has effect on the ECN and application functionality. The gravest is if the node drops packets with any ECN field values other than 00b. This can be detected by the receiver when it receives a RTCP SR packet indicating that a sender has sent a number of packets has not been received. The sender may also detect it based on the receivers RTCP RR packet where the extended sequence number is not advanced due to the failure to receive packets. If the packet loss is less than 100% then packet loss reporting in either the ECN feedback information or RTCP RR will indicate the situation. The other action is to re-mark a packet from ECT or CE to not-ECT. That has less dire results, however, it should be detected so that ECN usage can be suspended to prevent misusing the network.

The ECN feedback packet allows the sender to compare the number of ECT marked packets of different type with the number it actually sent. The number of ECT packets received plus the number of CE marked and lost packets should correspond to the number of sent ECT marked packets unless there is duplication in the network. If this number doesn't agree there are two likely reasons, a translator changing the stream or not carrying the ECN markings forward, or that some node re-marks the packets. In both cases the usage of ECN is broken on the path. By tracking all the different possible ECN field values a sender can quickly detect if some non-compliant behavior is happening on the path.

Thus packet losses and non-matching ECN field value statistics are possible indication of issues with using ECN over the path. The next section defines both sender and receiver reactions to these cases.

#### 6.4.1. Fallback mechanisms

Upon the detection of a potential failure both the sender and the receiver can react to mitigate the situation.

A receiver that detects a packet loss burst MAY schedule an early feedback packet to report this to the sender that includes at least the RTCP RR and the ECN feedback message. Thus speeding up the detection at the sender of the losses and thus triggering sender side

mitigation.

A sender that detects high packet loss rates for ECT-marked packets SHOULD immediately switch to sending packets as not-ECT to determine if the losses potentially are due to the ECT markings. If the losses disappear when the ECT-marking is discontinued, the RTP sender should go back to initiation procedures to attempt to verify the apparent loss of ECN capability of the used path. If a re-initiation fails then the two possible actions exist:

1. Periodically retry the ECN initiation to detect if a path change occurs to a path that is ECN capable.
2. Renegotiating the session to disable ECN support. This is a choice that is suitable if the impact of ECT probing on the media quality are noticeable. If multiple initiations has been successful but the following full usage of ECN has resulted in the fallback procedures then disabling of the ECN support is RECOMMENDED.

We foresee the possibility of flapping ECN capability due to several reasons: video switching MCU or similar middleboxes that selects to deliver media from the sender only intermittently; load balancing devices may in worst case result in that some packets take a different network path than the others; mobility solutions that switches underlying network path in a transparent way for the sender or receiver; and membership changes in a multicast group. It is however appropriate to mention that there are also issues such as re-routing of traffic due to a flappy route table or excessive reordering and other issues that are not directly ECN related but nevertheless cause problems in receivers.

#### 6.4.2. Interpretation of ECN Summary information

This section contains discussion on how you can use the ECN summary report information in detecting various types of ECN path issues. Lets start to review the information the reports provide on a per source (SSRC) basis:

**CE Counter:** The number of RTP packets received so far in the session with an ECN field set to CE (11b).

**ECT (0/1) Counters:** The number of RTP packets received so far in the session with an ECN field set to ECT (0) and ECT (1) respectively (10b / 01b).

not-ECT Counter: The number of RTP packets received so far in the session with an ECN field set to not-ECT (00b)

Lost Packets counter: The number of RTP packets that are expected minus the number received.

Extended Highest Sequence number: The highest sequence number seen when sending this report, but with additional bits, to handle disambiguation when wrapping the RTP sequence number field.

The counters will be initiated to zero to provide value for the RTP stream sender from the very first report. After the first report the changes between the latest received and the previous one is determined by simply taking the values of the latest minus the previous one, taking field wrapping into account. This definition is also robust to packet losses, since if one report is missing, the reporting interval becomes longer, but is otherwise equally valid.

In a perfect world the number of not-ECT packets received should be equal to the number sent minus the lost packets counter, and the sum of the ECT(0), ECT(1), and CE counters should be equal to the number of ECT marked packet sent. Two issues may cause a mismatch in these statistics: severe network congestion or unresponsive congestion control might cause some ECT-marked packets to be lost, and packet duplication might result in some packets being received, and counted in the statistics, multiple times (potentially with a different ECN-mark on each copy of the duplicate).

The level of packet duplication included in the report can be estimated from the sum over all of fields counting received packets compared to the number of packets sent. A high level of packet duplication increases the uncertainty in the statistics, making it more difficult to draw firm conclusions about the behaviour of the network. This issue is also present with standard RTCP reception reports.

Detecting clearing of ECN field: If the ratio between ECT and not-ECT transmitted in the reports has become all not-ECT or substantially changed towards not-ECT then this is clearly indication that the path results in clearing of the ECT field.

Dropping of ECT packets: To determine if the packet drop ratio is different between not-ECT and ECT marked transmission requires a mix of transmitted traffic. The sender should compare if the delivery percentage (delivered / transmitted) between ECT and not-ECT is significantly different. Care must be taken if the number of packets are low in either of the categories. One must also take into account the level of CE marking. A CE marked packet would have been dropped

unless it was ECT marked. Thus, the packet loss level for not-ECT should be approximately equal to the loss rate for ECT when counting the CE marked packets as lost ones. A sender performing this calculation needs to ensure that the difference is statistically significant.

If erroneous behavior is detected, it should be logged to enable follow up and statistics gathering.

## 7. Processing RTCP ECN Feedback in RTP Translators and Mixers

RTP translators and mixers that support ECN feedback are required to process, and potentially modify or generate, RTCP packets for the translated and/or mixed streams. This includes both downstream RTCP reports generated by the media sender, and also reports generated by the receivers, flowing upstream back towards the sender.

### 7.1. Fragmentation and Reassembly in Translators

An RTP translator may fragment or reassemble RTP data packets without changing the media encoding, and without reference to the congestion state of the networks it bridges. An example of this might be to combine packets of a voice-over-IP stream coded with one 20ms frame per RTP packet into new RTP packets with two 20ms frames per packet, thereby reducing the header overheads and so stream bandwidth, at the expense of an increase in latency. If multiple data packets are re-encoded into one, or vice versa, the RTP translator MUST assign new sequence numbers to the outgoing packets. Losses in the incoming RTP packet stream may also induce corresponding gaps in the outgoing RTP sequence numbers. An RTP translator MUST rewrite RTCP packets to make the corresponding changes to their sequence numbers, and to reflect the impact of the fragmentation or reassembly. This section describes how that rewriting is to be done for RTCP ECN feedback packets. Section 7.2 of [RFC3550] describes general procedures for other RTCP packet types.

RTCP ECN feedback packets (Section 5.1) contain six fields that are rewritten in an RTP translator that fragments or reassembles packets: the extended highest sequence number, the lost packets counter, the CE counter, and not-ECT counter, the ECT(0) counter, and the ECT(1) counter. The RTCP XR report block for ECN summary information (Section 5.2) includes a subset of these fields excluding the extended highest sequence number and lost packets counter. The procedures for rewriting these fields are the same for both types of RTCP ECN feedback packet.

When receiving an RTCP ECN feedback packet for the translated stream,



an RTP translator first determines the range of packets to which the report corresponds. The extended highest sequence number in the RTCP ECN feedback packet (or in the RTCP SR/RR packet contained within the compound packet, in the case of RTCP XR ECN summary reports) specifies the end sequence number of the range. For the first RTCP ECN feedback packet received, the initial extended sequence number of the range may be determined by subtracting the sum of the lost packets counter, the CE counter, the not-ECT counter, the ECT(0) counter and the ECT(1) counter from the extended highest sequence number (this will be inaccurate if there is packet duplication). For subsequent RTCP ECN feedback packets, the starting sequence number may be determined as being one after the extended highest sequence number of the previous RTCP ECN feedback packet received from the same SSRC. These values are in the sequence number space of the translated packets.

Based on its knowledge of the translation process, the translator determines the sequence number range for the corresponding original, pre-translation, packets. The extended highest sequence number in the RTCP ECN feedback packet is rewritten to match the final sequence number in the pre-translation sequence number range.

The translator then determines the ratio,  $R$ , of the number of packets in the translated sequence number space ( $\text{numTrans}$ ) to the number of packets in the pre-translation sequence number space ( $\text{numOrig}$ ) such that  $R = \text{numTrans} / \text{numOrig}$ . The counter values in the RTCP ECN feedback report are then scaled by dividing each of them by  $R$ . For example, if the translation process combines two RTP packets into one, then  $\text{numOrig}$  will be twice  $\text{numTrans}$ , giving  $R=0.5$ , and the counters in the translated RTCP ECN feedback packet will be twice those in the original.

The ratio,  $R$ , may have a value that leads to non-integer multiples of the counters when translating the RTCP packet. For example, a VoIP translator that combines two adjacent RTP packets into one if they contain active speech data, but passes comfort noise packets unchanged, would have an  $R$  values of between 0.5 and 1.0 depending on the amount of active speech. Since the counter values in the translated RTCP report are integer values, rounding will be necessary in this case.

When rounding counter values in the translated RTCP packet, the translator should try to ensure that they sum to the number of RTP packets in the pre-translation sequence number space ( $\text{numOrig}$ ). The translator should also try to ensure that no non-zero counter is rounded to a zero value, since that will lose information that a particular type of event has occurred. It is recognised that it may be impossible to satisfy both of these constraints; in such cases, it

is better to ensure that no non-zero counter is mapped to a zero value, since this preserves congestion adaptation and helps the RTCP-based ECN initiation process.

It should be noted that scaling the RTCP counter values in this way is meaningful only on the assumption that the level of congestion in the network is related to the number of packets being sent. This is likely to be a reasonable assumption in the type of environment where RTP translators that fragment or reassemble packets are deployed, as their entire purpose is to change the number of packets being sent to adapt to known limitations of the network, but is not necessarily valid in general.

The rewritten RTCP ECN feedback report is sent from the other side of the translator to that which it arrived (as part of a compound RTCP packet containing other translated RTCP packets, where appropriate).

## 7.2. Generating RTCP ECN Feedback in Media Transcoders

An RTP translator that acts as a media transcoder cannot directly forward RTCP packets corresponding to the transcoded stream, since those packets will relate to the non-transcoded stream, and will not be useful in relation to the transcoded RTP flow. Such a transcoder will need to interpose itself into the RTCP flow, acting as a proxy for the receiver to generate RTCP feedback in the direction of the sender relating to the pre-transcoded stream, and acting in place of the sender to generate RTCP relating to the transcoded stream, to be sent towards the receiver. This section describes how this proxying is to be done for RTCP ECN feedback packets. Section 7.2 of [RFC3550] describes general procedures for other RTCP packet types.

An RTP translator acting as a media transcoder in this manner does not have its own SSRC, and hence is not visible to other entities at the RTP layer. RTCP ECN feedback packets and RTCP XR report blocks for ECN summary information that are received from downstream relate to the translated stream, and so must be processed by the translator as if it were the original media source. These reports drive the congestion control loop and media adaptation between the translator and the downstream receiver. If there are multiple downstream receivers, a logically separate transcoder instance must be used for each receiver, and must process RTCP ECN feedback and summary reports independently to the other transcoder instances. An RTP translator acting as a media transcoder in this manner **MUST NOT** forward RTCP ECN feedback packets or RTCP XR ECN summary reports from downstream receivers in the upstream direction.

An RTP translator acting as a media transcoder will generate RTCP reports upstream towards the original media sender, based on the

reception quality of the original media stream at the translator. The translator will run a separate congestion control loop and media adaptation between itself and the media sender for each of its downstream receivers, and must generate RTCP ECN feedback packets and RTCP XR ECN summary reports for that congestion control loop using the SSRC of that downstream receiver.

### 7.3. Generating RTCP ECN Feedback in Mixers

An RTP mixer terminates one-or-more RTP flows, combines them into a single outgoing media stream, and transmits that new stream as a separate RTP flow. A mixer has its own SSRC, and is visible to other participants in the session at the RTP layer.

An ECN-aware RTP mixer must generate RTCP ECN feedback packets and RTCP XR report blocks for ECN summary information relating to the RTP flows it terminates, in exactly the same way it would if it were an RTP receiver. These reports form part of the congestion control loop between the mixer and the media senders generating the streams it is mixing. A separate control loop runs between each sender and the mixer.

An ECN-aware RTP mixer will negotiate and initiate the use of ECN on the mixed flows it generates, and will accept and process RTCP ECN feedback reports and RTCP XR report blocks for ECN relating to those mixed flows as if it were a standard media sender. A congestion control loop runs between the mixer and its receivers, driven in part by the ECN reports received.

An RTP mixer **MUST NOT** forward RTCP ECN feedback packets or RTCP XR ECN summary reports reports from downstream receivers in the upstream direction.

## 8. Implementation considerations

To allow the use of ECN with RTP over UDP, the RTP implementation must be able to set the ECT bits in outgoing UDP datagrams, and must be able to read the value of the ECT bits on received UDP datagrams. The standard Berkeley sockets API pre-dates the specification of ECN, and does not provide the functionality which is required for this mechanism to be used with UDP flows, making this specification difficult to implement portably.

## 9. IANA Considerations

Note to RFC Editor: please replace "RFC XXXX" below with the RFC

number of this memo, and remove this note.

#### 9.1. SDP Attribute Registration

Following the guidelines in [RFC4566], the IANA is requested to register one new SDP attribute:

- o Contact name, email address and telephone number: Authors of RFCXXXX
- o Attribute-name: ecn-capable-rtp
- o Type of attribute: media-level
- o Subject to charset: no

This attribute defines the ability to negotiate the use of ECT (ECN capable transport). This attribute should be put in the SDP offer if the offering party wishes to receive an ECT flow. The answering party should include the attribute in the answer if it wish to receive an ECT flow. If the answerer does not include the attribute then ECT MUST be disabled in both directions.

#### 9.2. RTP/AVPF Transport Layer Feedback Message

The IANA is requested to register one new RTP/AVPF Transport Layer Feedback Message in the table of FMT values for RTPFB Payload Types [RFC4585] as defined in Section 5.1:

Name:	RTCP-ECN-FB
Long name:	RTCP ECN Feedback
Value:	6
Reference:	RFC XXXX

#### 9.3. RTCP XR Report blocks

The IANA is requested to register one new RTCP XR Block Type as defined in Section 5.2:

Block Type:	13
Name:	ECN Summary Report
Reference:	RFC XXXX

#### 9.4. STUN attribute

A new STUN [RFC5389] attribute in the Comprehension-optional range under IETF Review (0x0000 - 0x3FFF) is request to be assigned to the STUN attribute defined in Section 6.2.2. The STUN attribute registry

can currently be found at: <http://www.iana.org/assignments/stun-parameters/stun-parameters.xhtml>.

#### 9.5. ICE Option

A new ICE option "rtp+ecn" is registered in the non-existing registry which needs to be created.

### 10. Security Considerations

The usage of ECN with RTP over UDP as specified in this document has the following known security issues that needs to be considered.

External threats to the RTP and RTCP traffic:

**Denial of Service affecting RTCP:** For an attacker that can modify the traffic between the media sender and a receiver can achieve either of two things. 1. Report a lot of packets as being Congestion Experience marked, thus forcing the sender into a congestion response. 2. Ensure that the sender disable the usage of ECN by reporting failures to receive ECN by changing the counter fields. The Issue, can also be accomplished by injecting false RTCP packets to the media sender. Reporting a lot of CE marked traffic is likely the more efficient denial of service tool as that may likely force the application to use lowest possible bit-rates. The prevention against an external threat is to integrity protect the RTCP feedback information and authenticate the sender of it.

**Information leakage:** The ECN feedback mechanism exposes the receivers perceived packet loss, what packets it considers to be ECN-CE marked and its calculation of the ECN-none. This is mostly not considered sensitive information. If considered sensitive the RTCP feedback shall be encrypted.

**Changing the ECN bits** An on-path attacker that see the RTP packet flow from sender to receiver and who has the capability to change the packets can rewrite ECT into ECN-CE thus forcing the sender or receiver to take congestion control response. This denial of service against the media quality in the RTP session is impossible for an end-point to protect itself against. Only network infrastructure nodes can detect this illicit re-marking. It will be mitigated by turning off ECN, however, if the attacker can modify its response to drop packets the same vulnerability exist.

Denial of Service affecting the session set-up signalling: If an attacker can modify the session signalling it can prevent the usage of ECN by removing the signalling attributes used to indicate that the initiator is capable and willing to use ECN with RTP/UDP. This attack can be prevented by authentication and integrity protection of the signalling. We do note that any attacker that can modify the signalling has more interesting attacks they can perform than prevent the usage of ECN, like inserting itself as a middleman in the media flows enabling wire-tapping also for an off-path attacker.

The following are threats that exist from misbehaving senders or receivers:

Receivers cheating A receiver may attempt to cheat and fail to report reception of ECN-CE marked packets. The benefit for a receiver cheating in its reporting would be to get an unfair bit-rate share across the resource bottleneck. It is far from certain that a receiver would be able to get a significant larger share of the resources. That assumes a high enough level of aggregation that there are flows to acquire shares from. The risk of cheating is that failure to react to congestion results in packet loss and increased path delay.

Receivers misbehaving: A receiver may prevent the usage of ECN in an RTP session by reporting itself as non ECN capable. Thus forcing the sender to turn off usage of ECN. In a point-to-point scenario there is little incentive to do this as it will only affect the receiver. Thus failing to utilise an optimisation. For multi-party session there exist some motivation why a receiver would misbehave as it can prevent also the other receivers from using ECN. As an insider into the session it is difficult to determine if a receiver is misbehaving or simply incapable, making it basically impossible in the incremental deployment phase of ECN for RTP usage to determine this. If additional information about the receivers and the network is known it might be possible to deduce that a receiver is misbehaving. If it can be determined that a receiver is misbehaving, the only response is to exclude it from the RTP session and ensure that it doesn't any longer have any valid security context to affect the session.

Misbehaving Senders: The enabling of ECN gives the media packets a higher degree of probability to reach the receiver compared to not-ECT marked ones on a ECN capable path. However, this is no magic bullet and failure to react to congestion will most likely only slightly delay a buffer under-run, in which its session also will experience packet loss and increased delay. There are some chance that the media senders traffic will push other traffic out

of the way without being effected to negatively. However, we do note that a media sender still needs to implement congestion control functions to prevent the media from being badly affected by congestion events. Thus the misbehaving sender is getting a unfair share. This can only be detected and potentially prevented by network monitoring and administrative entities. See Section 7 of [RFC3168] for more discussion of this issue.

We note that the end-point security functions needs to prevent an external attacker from affecting the solution easily are source authentication and integrity protection. To prevent what information leakage there can be from the feedback encryption of the RTCP is also needed. For RTP there exist multiple solutions possible depending on the application context. Secure RTP (SRTP) [RFC3711] does satisfy the requirement to protect this mechanism despite only providing authentication if a entity is within the security context or not. IPsec [RFC4301] and DTLS [RFC4347] can also provide the necessary security functions.

The signalling protocols used to initiate an RTP session also needs to be source authenticated and integrity protected to prevent an external attacker from modifying any signalling. Here an appropriate mechanism to protect the used signalling needs to be used. For SIP/SDP ideally S/MIME [RFC5751] would be used. However, with the limited deployment a minimal mitigation strategy is to require use of SIPS (SIP over TLS) [RFC3261] [RFC5630] to at least accomplish hop-by-hop protection.

We do note that certain mitigation methods will require network functions.

## 11. Examples of SDP Signalling

(tbd)

## 12. Open Issues

As this draft is under development some known open issues exist and are collected here. Please consider them and provide input.

1. The negotiation and directionality attribute is going to need some consideration for multi-party sessions when readonly capability might be sufficient to enable ECN for all incoming streams. However, it would be beneficial to know if no potential sender support setting ECN.

2. Consider initiation optimizations that allows for multi SSRC sender nodes to still have rapid usage of ECN.
3. Should we report congestion in bytes or packets? RTCP usually does this in terms of packets, but there may be an argument that we want to report bytes for ECN.  
draft-ietf-tsvwg-byte-pkt-congest is extremely unclear on what is the right approach. (csp)
4. Add examples of SDP signalling

### 13. References

#### 13.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [RFC3611] Friedman, T., Caceres, R., and A. Clark, "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, November 2003.
- [RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.
- [RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.
- [RFC5348] Floyd, S., Handley, M., Padhye, J., and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 5348, September 2008.
- [RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.



## 13.2. Informative References

- [I-D.ietf-avt-rtp-no-op]  
Andreasen, F., "A No-Op Payload Format for RTP",  
draft-ietf-avt-rtp-no-op-04 (work in progress), May 2007.
- [I-D.zimmermann-avt-zrtp]  
Zimmermann, P., Johnston, A., and J. Callas, "ZRTP: Media  
Path Key Agreement for Unicast Secure RTP",  
draft-zimmermann-avt-zrtp-22 (work in progress),  
June 2010.
- [RFC2974] Handley, M., Perkins, C., and E. Whelan, "Session  
Announcement Protocol", RFC 2974, October 2000.
- [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.
- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model  
with Session Description Protocol (SDP)", RFC 3264,  
June 2002.
- [RFC3540] Spring, N., Wetherall, D., and D. Ely, "Robust Explicit  
Congestion Notification (ECN) Signaling with Nonces",  
RFC 3540, June 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.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the  
Internet Protocol", RFC 4301, December 2005.
- [RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram  
Congestion Control Protocol (DCCP)", RFC 4340, March 2006.
- [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.

- [RFC4585] 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.
- [RFC4960] Stewart, R., "Stream Control Transmission Protocol", RFC 4960, September 2007.
- [RFC5124] Ott, J. and E. Carrara, "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)", RFC 5124, February 2008.
- [RFC5506] Johansson, I. and M. Westerlund, "Support for Reduced-Size Real-Time Transport Control Protocol (RTCP): Opportunities and Consequences", RFC 5506, April 2009.
- [RFC5630] Audet, F., "The Use of the SIPS URI Scheme in the Session Initiation Protocol (SIP)", RFC 5630, October 2009.
- [RFC5751] Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", RFC 5751, January 2010.
- [RFC5760] Ott, J., Chesterfield, J., and E. Schooler, "RTP Control Protocol (RTCP) Extensions for Single-Source Multicast Sessions with Unicast Feedback", RFC 5760, February 2010.

#### Authors' Addresses

Magnus Westerlund  
Ericsson  
Farogatan 6  
SE-164 80 Kista  
Sweden

Phone: +46 10 714 82 87  
Email: magnus.westerlund@ericsson.com

Ingemar Johansson  
Ericsson  
Laboratoriegrand 11  
SE-971 28 Lulea  
SWEDEN

Phone: +46 73 0783289  
Email: [ingemar.s.johansson@ericsson.com](mailto:ingemar.s.johansson@ericsson.com)

Colin Perkins  
University of Glasgow  
School of Computing Science  
Glasgow G12 8QQ  
United Kingdom

Email: [csp@csp Perkins.org](mailto:csp@csp Perkins.org)

Piers O'Hanlon  
University College London  
Computer Science Department  
Gower Street  
London WC1E 6BT  
United Kingdom

Email: [p.ohanlon@cs.ucl.ac.uk](mailto:p.ohanlon@cs.ucl.ac.uk)

Ken Carlberg  
G11  
1600 Clarendon Blvd  
Arlington VA  
USA

Email: [carlberg@g11.org.uk](mailto:carlberg@g11.org.uk)



Audio/Video Transport WG  
Internet-Draft  
Updates: 3984bis  
(if approved)  
Intended status: Standards Track  
Expires: April 28, 2011

T. Kristensen  
M. Walters  
Cisco  
October 25, 2010

Additional H.241 Parameter in the RTP Payload Format for H.264 Video  
draft-kristensen-avt-rtp-h241param-01

Abstract

As systems increasingly are able to run video at 60 frames per second, there is a need to signal desired frame rate to optimise resolution choices and to avoid unnecessary resource or energy usage. This document defines a new optional parameter addressing recent extensions currently supported in H.323 systems: The signalling of the maximum frames per second that can be efficiently received or that can be sent.

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 April 28, 2011.

Copyright Notice

Copyright (c) 2010 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 and Overview . . . . .	3
1.1. New Parameter max-fps . . . . .	3
1.2. Existing Schemes . . . . .	4
2. Terminology . . . . .	4
3. Payload Format Parameters . . . . .	4
3.1. Media Type Registration . . . . .	5
4. SDP Parameters . . . . .	5
4.1. Mapping of the optional parameter to SDP . . . . .	5
4.2. Usage with SDP offer/answer . . . . .	5
4.3. Usage in Declarative Session Descriptions . . . . .	6
4.4. Examples . . . . .	6
5. Co-existence . . . . .	6
6. IANA Considerations . . . . .	6
7. Security Considerations . . . . .	6
8. Open issues . . . . .	7
9. Acknowledgements . . . . .	7
10. References . . . . .	7
10.1. Normative References . . . . .	7
10.2. Informative references . . . . .	8
Appendix A. Change History . . . . .	8
A.1. -00 to -01 . . . . .	8
Appendix B. Figures . . . . .	8
Authors' Addresses . . . . .	12

## 1. Introduction and Overview

The extended video procedures and control signals for H.300 series terminals in ITU-T H.241 [ITU.H241.2006], associated with the ITU-T H.264 [ITU.H264.2007] codec, continue to evolve. This document describes a new parameter specified in H.241 - Amendment 2 [ITU.H241Amd2.2009] used to indicate the maximum picture rate that can be efficiently received or the maximum picture rate that can be sent. This proposal defines a media type parameter for maximum picture rate signalling and allows use in SDP-based systems, as this is not covered in RFC3984bis [I-D.ietf-avt-rtp-rfc3984bis].

Editorial note: This draft may be used to specify further, new H.241 parameters if applicable and when new parameters are defined.

### 1.1. New Parameter max-fps

The use of MaxMBPS and MaxFS, either signalled implicitly via supported level or as the optional parameters max-mbps and max-fs [I-D.ietf-avt-rtp-rfc3984bis], underspecify the video stream and can lead to wasted bandwidth and processing resources, as well as unnecessary energy consumption and, where applicable, shorter battery duration.

For example, a decoder signalling support for w1080p30 (243000 MBPS and 8100 MB) implicitly allows the sending of w720p60 (216000 MBPS and 3600 MB). Similarly, if a decoder signalled support for w720p30 (108000 MBPS and 3600 MB) the encoder could choose to send w480p60, w448p60 or even wCIF at 120 frames per second. Graphs visualizing these examples are included in Appendix B, where the existing parameters are used, with the addition of max-fps to constrain the allowed parameter range as preferred.

If the encoder chooses to send at a higher frame rate than preferred by the receiver side, the decoder will normally discard the additional frames after decoding them. The transmission of the extra frames and the processing of frames to just discard them are wasteful and the bandwidth and processing could be used more effectively.

The new parameter MaxFPS is introduced in H.241 - Amendment 2 to remove this issue, by constraining the allowed parameter range for systems supporting this extension. This draft proposes adding the optional parameter max-fps for use in SDP-based signalling.

The specification of max-fps in this draft will also ease the gatewaying between H.323 and SIP systems by making the translation straight-forward for this parameter as well.

It is worth emphasizing that systems that does not support and understand the max-fps parameter are free to produce streams as before, within the bounds specified by the existing parameters.

## 1.2. Existing Schemes

A general attribute "a=framerate" is specified in SDP [RFC4566] as a recommendation for the encoding of video. It is a media-level attribute and is defined only for video. The "a=framerate" attribute applies to all video payloads in the media section it belongs. This may work well in scenarios where the maximum frame rate is identical for all the video codecs present in the video media section. This is not always the case.

In addition to the general "a=framerate" attribute, some video standards have some sort of frame rate signalling in their SDP description. For instance, in the H.263 and H.261 RTP payload format specifications [RFC4629][RFC4587] the maximum frame rate capability is signalled as an integer value Minimum Picture Interval (MPI). This MPI value assumes an upper bound of 30 frames per second. Another example is the VC-1 video codec [RFC4425], where an optional parameter framerate is specified as an "a=fmtp" parameter.

As the examples above show, different media subtypes enables different means to signal the maximum frame rate capability. Some are explicit, others implicit. Some even have an implicit assumption of the maximum frame rate it is possible to express. Also, using only the SDP "a=framerate" attribute for all payloads associated with a media line is not feasible in general. Refer to Section 5 for recommended usage of "a=framerate" together with the max-fps parameter specified in this draft.

## 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].

In video formats based on NTSC, 30 frames per second and 60 frames per second are used repeatedly. However, the actual values are  $30/1.001=29.97$  and  $60/1.001=59.94$ . Values derived from these fractional values MUST be used in signalling descriptions.

## 3. Payload Format Parameters

The optional H264 media subtype parameter max-fps specified below



comes in addition to the list of optional H264 media subtype parameters defined in Section 8.1 of RFC3984bis [I-D.ietf-avt-rtp-rfc3984bis].

### 3.1. Media Type Registration

New OPTIONAL parameter:

**max-fps:** This parameter SHOULD be used to indicate the maximum picture rate that can be efficiently received or the maximum picture rate that can be sent. The value of max-fps is an integer in units of hundredths of frames per second. Note that this is not necessarily the frame rate at which the maximum frame size can be sent. The encoder SHOULD use a frame rate equal to or less than this value. However, encoders MAY choose to send a higher frame rate. If the parameter is absent then the encoder is free to choose any frame rate.

## 4. SDP Parameters

### 4.1. Mapping of the optional parameter to SDP

The optional parameter max-fps, when present, MUST be included in the "a=fmtp" line of SDP. The parameters in the "a=fmtp" line are expressed as a media type string, in the form of a semicolon separated list of parameter=value pairs.

Editorial note: To be decided whether the stream property (sender capability) usage of max-fps is useful for SDP usage, i.e. with sendonly and for declarative usage.

### 4.2. Usage with SDP offer/answer

When H.264 is offered over RTP using SDP in an offer/answer exchange [RFC3264] for negotiation of unicast streams, the following limitations and rules applies:

The interpretation of the optional parameter max-fps depends on the direction attribute. When the direction attribute is sendonly, then it indicates the maximum picture rate that shall be sent. When the direction attribute is sendrecv or recvonly, the value of this parameter indicates the maximum picture frame rate that the receiver can efficiently handle. Any encoder that understands the parameter semantics shall constrain the frame rate to rates up to that specified. A receiver should have the ability to process video from a sender that does not understand this parameter.

The profile-level-id parameter MUST be present in the same receiver capability description that contains this parameter.

#### 4.3. Usage in Declarative Session Descriptions

When H.264 over RTP is offered with SDP in a declarative style, the max-fps parameter is used to declare the actual stream property and the value indicates the maximum picture rate that shall be sent.

#### 4.4. Examples

Two simple examples of SDP descriptions with the max-fps parameter. A NTSC-based system supporting maximum 30 frames per second could offer:

```
m=video 49170 RTP/AVP 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E; max-fps=2997
```

A description for a system supporting a true 60 frames per second:

```
m=video 49170 RTP/AVP 99
a=rtpmap:99 H264/90000
a=fmtp:99 profile-level-id=42A01E; max-fps=6000
```

#### 5. Co-existence

The "a=framerate" limitation allows a global maximum frame rate to be specified for the video media, this limit applies to all codecs in the video media section. In cases where both "a=framerate" and codec specific limits are specified, the maximum frame rate that can be used for a codec is the minimum of the values specified globally and specifically for a codec.

#### 6. IANA Considerations

This draft updates RFC3984bis by adding a new optional parameter to the H264 media subtype. The parameter is specified in Section 3 of this document.

#### 7. Security Considerations

No separate considerations introduced in this document. Refer to section 9 of RFC3984bis [I-D.ietf-avt-rtp-rfc3984bis].

## 8. Open issues

The open issues are marked as editorial notes in the draft. An overview of the issues:

Placeholder: This draft may be used for new H.241 parameters if applicable. Depends on the timeline of this draft and proposals for H.241 extensions. See Section 1.

Stream property: Specification for max-fps as stream property might be removed, if not feasible or actually required. Pending. See Section 4.1.

## 9. Acknowledgements

The authors would like to acknowledge Mo Zanaty (mzanaty@cisco.com) and Roni Even (even.roni@huawei.com) for helpful input and comments.

## 10. References

### 10.1. Normative References

[I-D.ietf-avt-rtp-rfc3984bis]

Wang, Y., Even, R., Kristensen, T., and R. Jesup, "RTP Payload Format for H.264 Video", draft-ietf-avt-rtp-rfc3984bis-12 (work in progress), October 2010.

[ITU.H241.2006]

International Telecommunications Union, "Extended video procedures and control signals for H.300-series terminals", ITU-T Recommendation H.241, May 2006.

[ITU.H241Amd2.2009]

International Telecommunications Union, "Extended video procedures and control signals for H.300-series terminals", ITU-T Recommendation H.241 - Amendment 2, December 2009.

[ITU.H264.2007]

International Telecommunications Union, "Advanced video coding for generic audiovisual services", ITU-T Recommendation H.264, November 2007.

[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.

## 10.2. Informative references

- [RFC4425] Klemets, A., "RTP Payload Format for Video Codec 1 (VC-1)", RFC 4425, February 2006.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.
- [RFC4587] Even, R., "RTP Payload Format for H.261 Video Streams", RFC 4587, August 2006.
- [RFC4629] Ott, H., Bormann, C., Sullivan, G., Wenger, S., and R. Even, "RTP Payload Format for ITU-T Rec. H.263 Video", RFC 4629, January 2007.

## Appendix A. Change History

### A.1. -00 to -01

1. Tightened text describing "a=maxprate" and H.26x frame rate signalling parameters, to get the real message clearer through.
2. Emphasized the optimization and energy consumption/battery duration as reasons for specifying max-pfs.
3. Added graphs as an aid visualizing the effect of max-fps, together with macroblock and frame size signalling
4. Clarified the 60/30 and 29.97/59.94 value usage in the document.
5. A better text regarding co-existence with "a=framerate" added.
6. Misc. editorial fixes and cleanup.

## Appendix B. Figures

Graphical representation of the usage of the existing parameters for signalling supported number of macroblocks per second (MaxMBPS) and frame size (MaxFS), with addition of the proposed max-fps parameter in an example scenario. Figure 1 assumes that an encoder can encode at one of three resolutions wCIF, w720p and w1080p, and three different frame rates 30, 60 and 120 frames per second (fps), but not all combinations of these are possible.

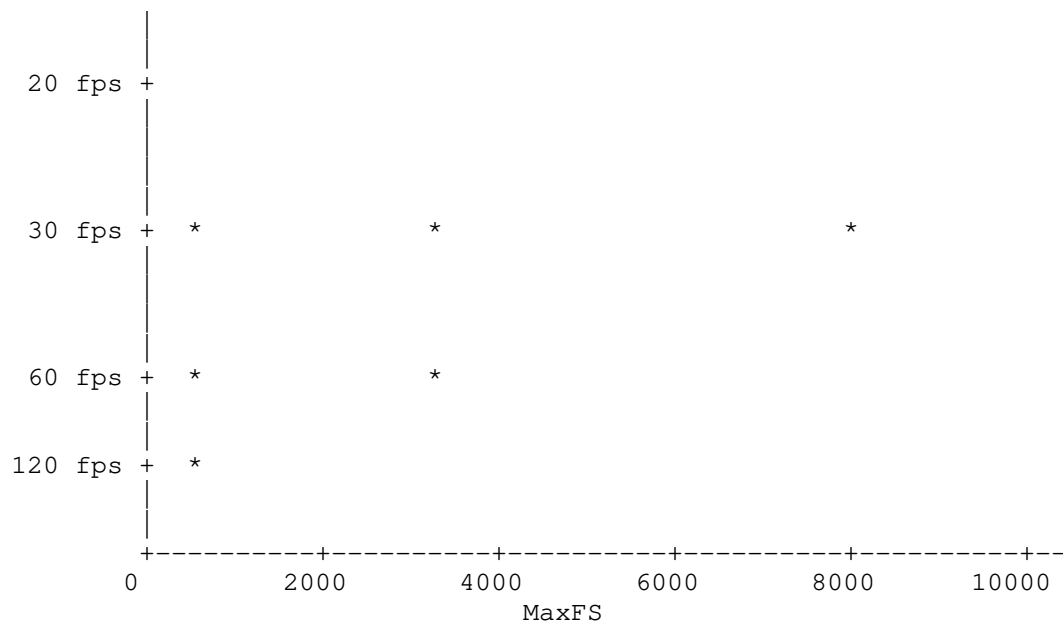


Figure 1: Encoder capabilities

The decoder can then signal its capabilities. For example, if it can decode 108000 macroblocks per second, then the encoder can choose any combination of frame size and frame rate within the dotted area as shown in Figure 2.

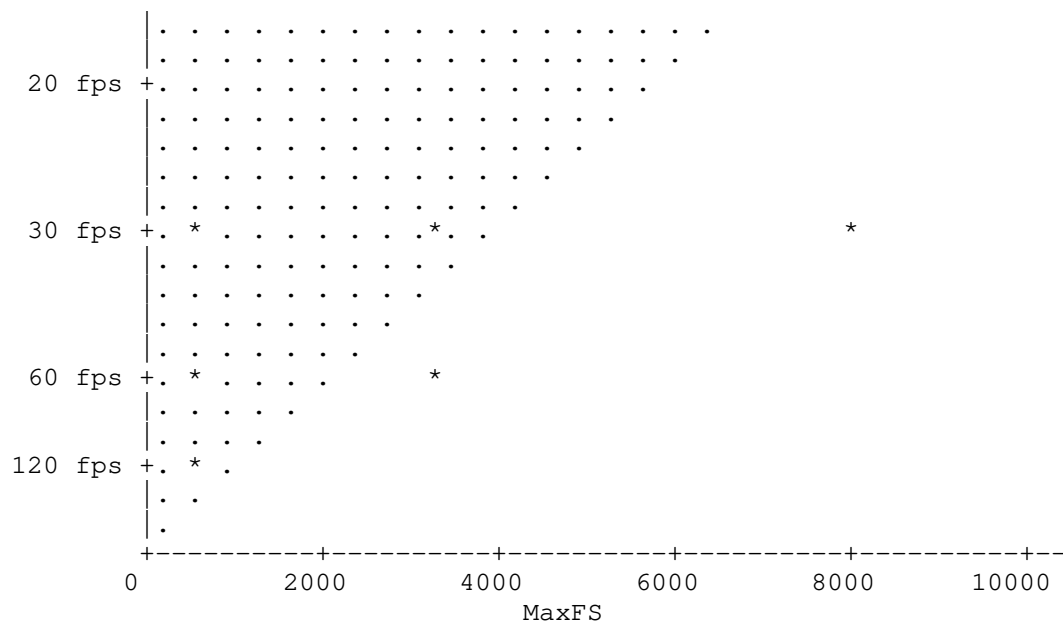


Figure 2: MaxMBPS indicated

If the decoder signals a maximum frame size (either explicitly as max-fs or via the supported level), then this further constrains the choice of the decoder as shown in Figure 3.

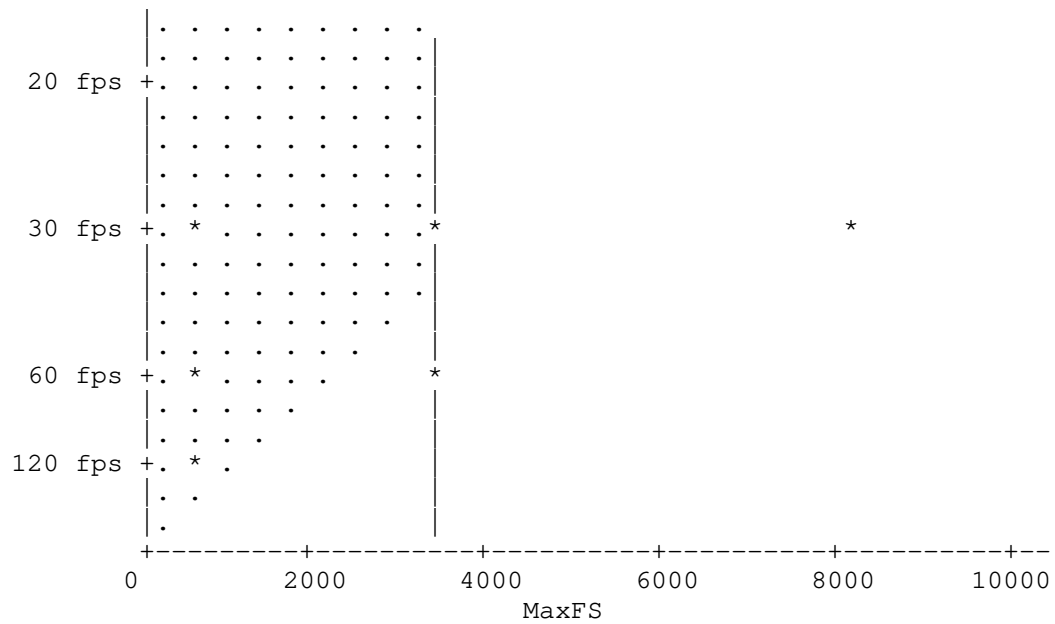


Figure 3: MaxMBPS and MaxFPS indicated

The parameter max-fps is added as a horizontal limit in Figure 4 restricting the allowed parameter space after the encoder's preferences.

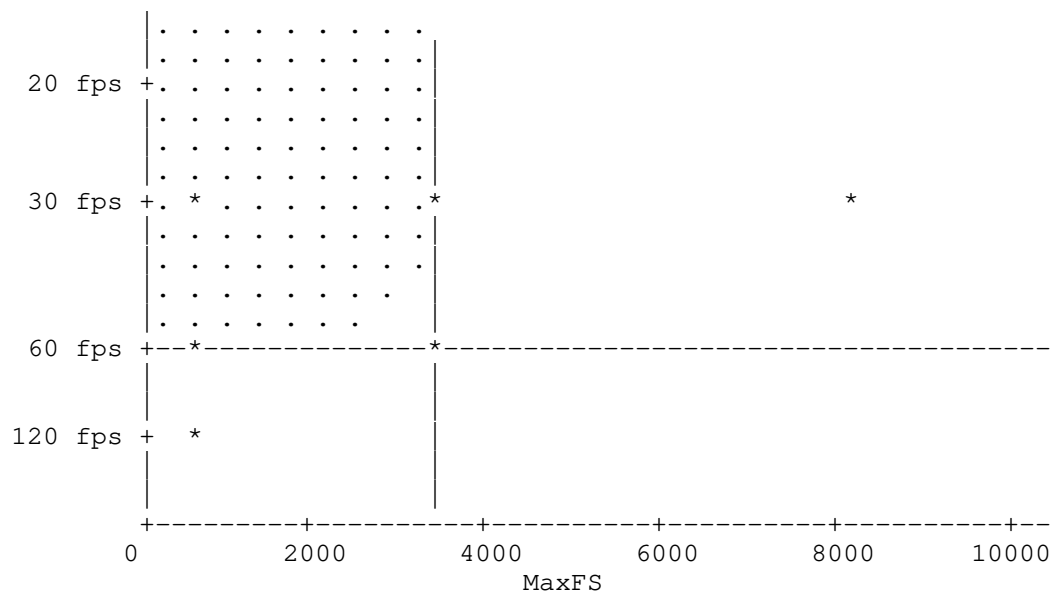


Figure 4: Adding max-fps to further constrain

## Authors' Addresses

Tom Kristensen

Cisco

Philip Pedersens vei 22

N-1366 Lysaker

Norway

Phone: +47 67125125

Email: tomkrist@cisco.com, tomkri@ifi.uio.no

Malcolm Walters

Cisco

14 Waterside Drive

Langley, Slough

UK

Email: malcowal@cisco.com





AVT Working Group  
Internet-Draft  
Intended status: Experimental  
Expires: April 28, 2011

V. Singh  
T. Karkkainen  
J. Ott  
S. Ahsan  
Aalto University  
L. Eggert  
Nokia  
October 25, 2010

Multipath RTP (MPRTP)  
draft-singh-avt-mprtp-01

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 April 28, 2011.

Copyright Notice

Copyright (c) 2010 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
1.3. Use cases . . . . .	4
2. Goals . . . . .	5
2.1. Functional goals . . . . .	5
2.2. Compatibility goals . . . . .	6
3. RTP Topologies . . . . .	6
4. MP RTP Architecture . . . . .	6
4.1. Relationship of MP RTP and Session Signaling . . . . .	8
5. Example Media Flow Diagrams . . . . .	8
5.1. Streaming Use Case . . . . .	8
5.2. Conversational Use Case . . . . .	9
5.3. Challenges with Multipath Interface Discovery . . . . .	10
6. MP RTP Functional Blocks . . . . .	10
7. Available Mechanisms Within the Functional Blocks . . . . .	11
7.1. Session Setup . . . . .	11
7.2. Expanding RTP . . . . .	11
7.3. Adding New Interfaces . . . . .	11
7.4. Expanding RTCP . . . . .	12
7.5. Checking and Failure Handling . . . . .	12
8. MP RTP Protocol . . . . .	12
8.1. MP RTP Session Establishment . . . . .	12
8.1.1. Subflow or Interface Advertisement . . . . .	13
8.1.2. Path selection . . . . .	14
8.1.3. Opening subflows . . . . .	14
8.2. Packet Transmission . . . . .	14
8.3. Playout Considerations at the Receiver . . . . .	15
8.4. Flow specific RTCP Statistics and RTCP Aggregation . . . . .	15
8.5. Packet Format . . . . .	15
8.5.1. MP RTP RTP Header Extension . . . . .	15
8.5.2. Interface Address Advertisement block . . . . .	17
8.5.3. MP RTP RTCP Header Extension . . . . .	18
9. SDP Considerations . . . . .	20
9.1. Increased Throughput . . . . .	20
10. Acknowledgements . . . . .	20
11. IANA Considerations . . . . .	21
12. Security Considerations . . . . .	21
13. References . . . . .	21
13.1. Normative References . . . . .	21
13.2. Informative References . . . . .	21
Authors' Addresses . . . . .	22

## 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 often there is 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 transmit over different paths. In effect, this pools the resource capacity of multiple paths.

Other IETF transport protocols that are capable of using multiple paths include SCTP [7], MPTCP [8] and SHIM6 [9]. However, these protocols are not suitable for realtime 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 connection.
- o Interface: A 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: A 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.

### 1.3. Use cases

The primary use case for MPRTP 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 one RTP stream over multiple paths can reduce the bandwidth usage, compared to transmitting the same stream along a single path. This reduces the impact on other traffic.
- 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 MP RTP 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 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

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

- o Application Compatibility: MPRTP service model MUST be backwards compatible with existing RTP applications, i.e., an MPRTP 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 MPRTP stack MAY provide extended APIs so that the application can configure any additional features provided by the MPRTP 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 [10] 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. MPRTP 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. MPRTP 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.

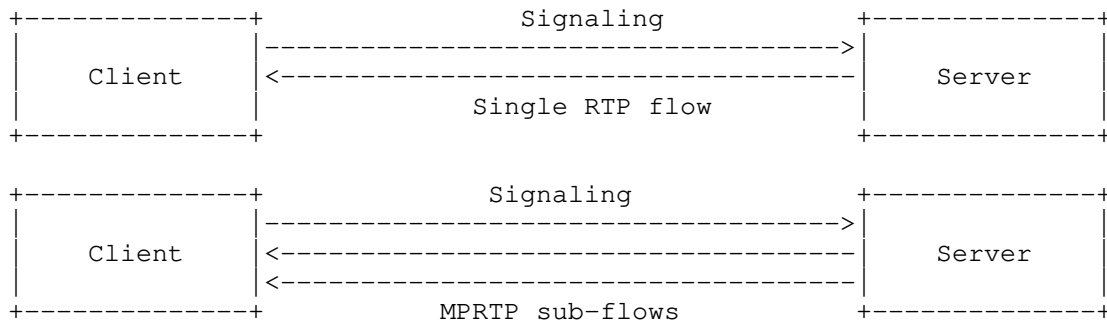


Figure 1: Comparison between traditional RTP streaming and MPRTTP

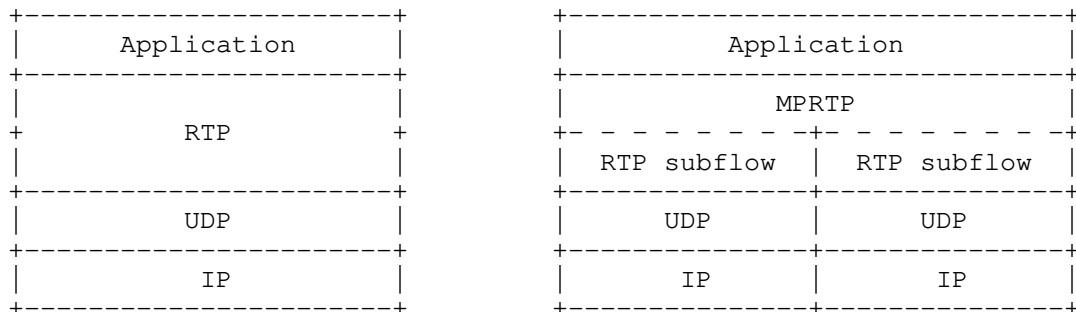


Figure 2: MPRTTP Architecture

Figure 2 illustrates the differences between the standard RTP stack and the MPRTTP stack. MPRTTP 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 MPRTTP layer performs the following functions:

- o **Path Management:** The layer is aware of alternate paths to the peer, which may, for example, be the peer's multiple interfaces to send 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 interface.
- 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.

#### 4.1. Relationship of MPRTTP and Session Signaling

Session signaling (e.g., SIP [11], RTSP [12]) SHOULD be done over failover-capable or multipath-capable transport for e.g., SCTP [7] or MPTCP [8] instead of TCP or UDP.

### 5. Example Media Flow Diagrams

There may be many complex technical scenarios for MPRTTP, however, this memo only considers the following two scenarios: 1) an 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 the endpoints are MPRTTP-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. Furthermore, host B splits the multimedia stream into two 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".

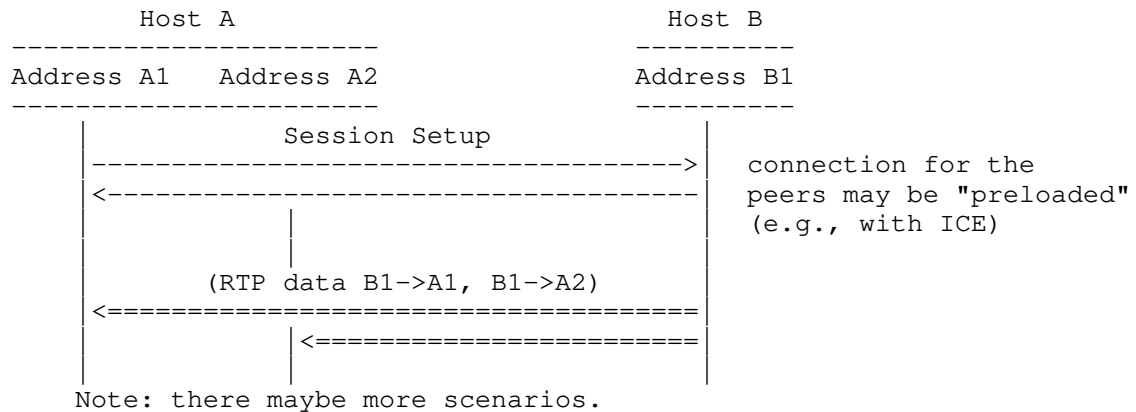


Figure 3: Unidirectional media flow

## 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. All interfaces that receive data send back RTCP receiver statistics for each path. The peers balance their own multimedia stream over multiple links based on the reception statistics from its peer and its own volume of traffic. Figure 4 describes an example of a bidirectional flow.

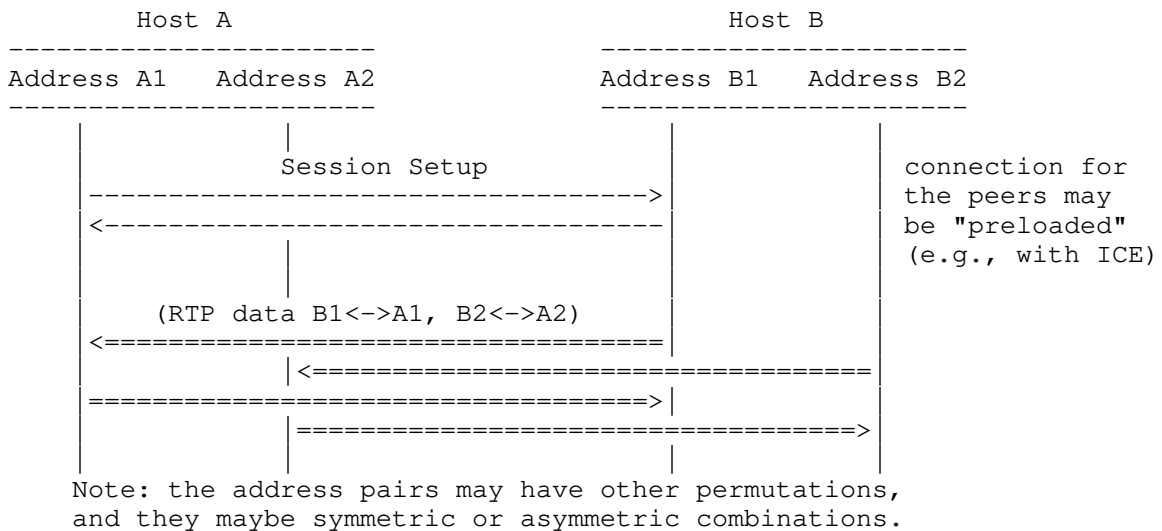


Figure 4: Bidirectional flow

### 5.3. Challenges with Multipath Interface Discovery

For some applications, where the user expects immediate playback, e.g., High Definition Media Streaming or IPTV, it may not be possible to perform connectivity checks within the given time bound. In these cases, connectivity checks MAY need to be done ahead of time.

[Editor: ICE or any other system would need to aware of the peer's interfaces ahead of time].

## 6. MPRTTP Functional Blocks

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

1. **Session Setup:** Multipath session setup is an upgrade or add-on to a typical RTP session. 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.
2. **Expanding RTP:** For a multipath session, each subflow MUST look like an independent RTP flow, so that individual RTCPs can be generated per subflow. Furthermore, MPRTTP splits the single multimedia stream into multiple subflows based on path characteristics and dynamically adjusts the load on each link.
3. **Adding Interfaces:** Interfaces on the host need to be regularly discovered and signaled. This can be done at the session setup and/or during the session. When discovering and receiving new interfaces, the MPRTTP layer needs to select address and port pairs.
4. **Expanding RTCP:** MPRTTP MUST recombine RTCP reports from each path to re-create a single RTCP message to maintain backward compatibility with legacy applications.
5. **Maintenance and Failure Handling:** In a multi-homed endpoint interfaces may appear and disappear. If this happens at the sender, it has to re-adjust the load on the available links. On the other hand, if this occurs on 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 receiver has to explicitly signal the disappearance of an interface, or the sender has to detect it. What happens if the

interface that setup the session disappears? does the control channel also failover? re-start the session?

6. Teardown: The MP RTP 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

MP 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., SDP[13] in SIP[11], RTSP [12]) or in-band signaling (e.g., RTP/RTCP header extension). Furthermore, ICE [3] may be used for discovering and performing connectivity checks during session setup.

### 7.2. Expanding RTP

RTCP [1] is generated per media session. However, with MP RTP, 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 MP RTP sender SHOULD add subflow-specific sequence numbers and RTP timestamps to help calculate fractional losses, jitter, RTT, playout time, etc., for each path. An example RTP header extension for MP RTP is shown in Section 8.5).

### 7.3. Adding New Interfaces

When interfaces appear and disappear mid-session, ICE [3] may be used for discovering interfaces and performing connectivity checks. However, MP RTP may require a capability re-negotiation (using SDP) to include all these new interfaces. This method is referred to as out-of-band multipath advertisement.

Alternatively, when new interfaces appear the interface advertisements may be done in-band using RTP/RTCP extensions. The peers perform connectivity checks (see Figure 5 for more details). If the connectivity packets are received by the peers, then multimedia data can flow between the new address, port pairs.

#### 7.4. Expanding RTCP

Multiple subflows in MP RTP affect RTCP bandwidth and RTCP reporting interval calculations. RTCP report scheduling for each subflow may cause a problem for RTCP recombined and reconstruction in cases when 1) RTCP for a subflow is lost, and 2) RTCP for a subflow arrives slower than other subflows. (There maybe other cases as well.)

The subflow RTCP RR reports at the sender help balance the load along each path. However, this document doesn't cover algorithms for congestion control or load balancing.

#### 7.5. Checking and Failure Handling

[Editor:If the original interface that setup the session disappears then does the session signaling failover to another interface? Can we recommend that SIP/RTSP be run over MPTCP, SCTP].

### 8. MP RTP Protocol

To provide a more concrete basis for discussion, in this section we illustrate a solution. To enable a quick start to a multimedia session, we presume that a multipath session SHOULD be upgraded from a single path session. Therefore, no explicit changes are needed in multimedia session setup and the session can be setup as before.

#### 8.1. MP RTP Session Establishment

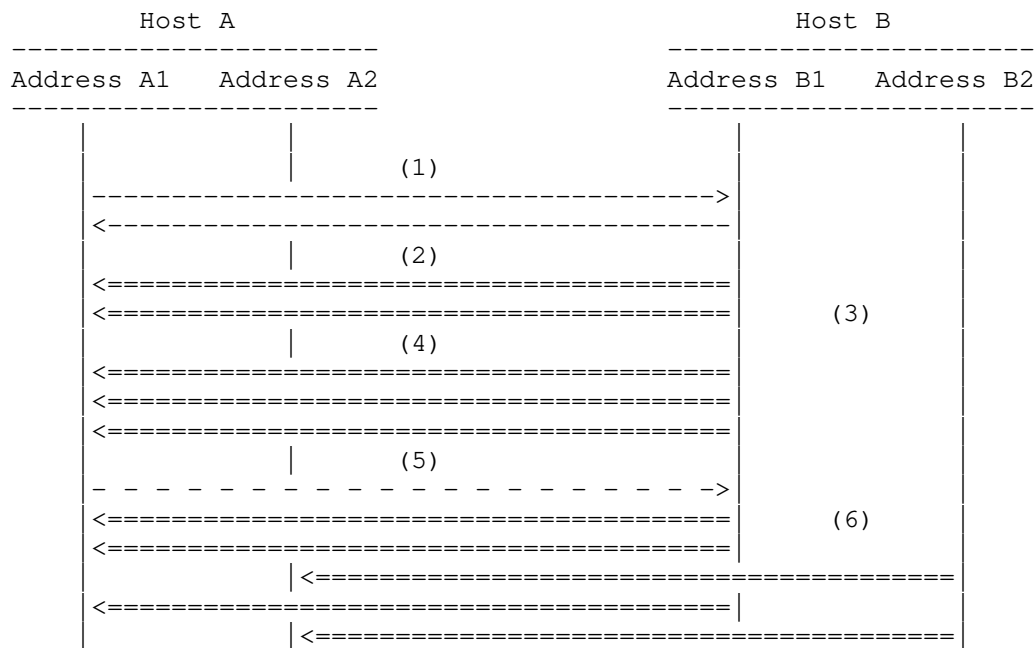
Initially, the session is set up as a standard single path multimedia session. The bullet points below explain the different steps shown in Figure 5.

- (1) The first two interactions between the hosts describes the standard session setup. This may be SIP or RTSP.
- (2) Following the setup, like in a conventional RTP scenario, host B using interface B1 starts to stream data to host A at interface A1.
- (3) Host B is an MP RTP-capable media sender and becomes aware of another interface B2.
- (4) Host B advertises the multiple interface addresses using an RTP header extensions.
- (5) Host A is an MP RTP-capable media receiver and becomes aware of another interface A2. It advertises the multiple interface

addresses using an RTCP RR extension.

Side note, if the MP RTP-capable hosts have no additional interfaces, then the hosts SHOULD still advertise a single interface.

(6) Each hosts receives information about the additional interfaces and perform connectivity tests (not shown in figure) and if the paths are reachable then the hosts start to stream the multimedia content using the additional paths.



Key:

| Interface

---> Signaling Protocol

<=== RTP Packets

- -> RTCP Packet

Figure 5: MP RTP New Interface

#### 8.1.1.1. Subflow or Interface Advertisement

MP RTP-capable media senders SHOULD use the RTP header extension defined in Figure 7 to advertise their interfaces. Each unique address is encapsulated in a Interface Advertisement block and

contains the IP address, RTP port, and RTCP port addresses. The Interface Advertisement blocks are ordered based on decreasing priority level.

On receiving the MPRTP Interface Advertisement, the receiver will either ignore the RTP header extension if it is not MPRTP capable or MUST respond with its own set of interfaces in decreasing order of priority. If the sender and receiver are MPRTP-capable but have only one interface, then they MUST respond with the default interface address, RTP and RTCP port addresses. If the sender receives an RTCP report without the MPRTP RTCP block after advertising its interfaces, then the sender MUST presume that the receiver is not MPRTP capable. Figure 9 illustrates an RTCP format for MPRTP Interface Advertisement.

#### 8.1.2. Path selection

After MPRTP support has been discovered and interface advertisements have been exchanged, the sender MUST initiate connectivity checks to determine which interface pairs offer valid paths between the sender and the receiver. To initiate a connectivity check, the sender sends an RTP packet with MPRTP extension header with MPR\_Type = 0x02 and no RTP payload. The receiver replies with an MPRTP RTCP packet with type MPRR\_Type = 0x02. After the sender receives the reply, the path is considered a valid candidate for subflow establishment.

The sender MAY choose to do any number of connectivity checks for any interface pairs at any point in a session.

#### 8.1.3. Opening subflows

The sender may open any number of subflows after performing connectivity checks. MPRTP MUST associate a Flow ID to each subflow. To open a new subflow, the sender simply starts sending the RTP packets with an MPRTP extension shown in Figure 6. The MPRTP extension carries a mapping of a subflow packet to the aggregate flow. Namely, sequence numbers and timestamps associated to the subflow.

#### 8.2. Packet Transmission

The MPRTP layer SHOULD associate an RTP packet to 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 changing path characteristics, an MPRTP sender might change its scheduling strategy during an ongoing session. The MPRTP sender MUST also populate the flow specific fields described in the MPRTP extension

header (see Section 8.5.1).

### 8.3. Playout Considerations at the Receiver

A media receiver, irrespective of MPRTTP support or not, should be able to playback the media stream because the received RTP packets are compliant to [1], i.e., a non-MPRTTP receiver will ignore the MPRTTP 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 (adaptive playout) of the received RTP packets.

### 8.4. Flow specific RTCP Statistics and RTCP Aggregation

The aggregate RTCP report may not provide sufficient per path information to an MPRTTP scheduler. Specifically, the scheduler should be aware of each path's RTT, which an aggregate RTCP cannot provide.

[Editor: 1) Should the RTCP RRs sent per path carry a) the aggregate and the path's RR or b) the aggregate and RR of each path.

2) Should the per path RTCP Interval be dependent on the overall session bitrate or per path interval receiver rate?]

### 8.5. Packet Format

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

#### 8.5.1. MPRTTP RTP Header Extension

The MPRTTP header extension is used 1) to pack single stream RTP data into multiple subflows, 2) to advertise the multiple interface addresses for a media sender, and 3) perform connectivity check on the new interfaces.

MPRTTP RTP header extension for a subflow:



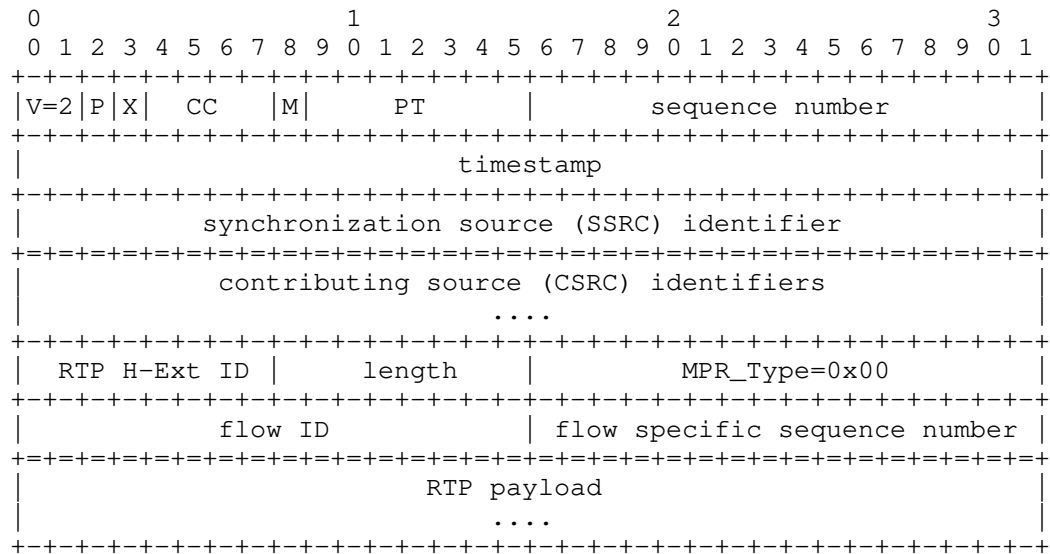


Figure 6: MPRTTP header for subflow

RTP H-Ext ID and length: 8-bits each

It conforms to the 2-byte RTP header extension defined in [4].

RTP H-Ext=TBD

The 8-bit length field is the length of extension data in bytes not including the RTP H-Ext ID and length fields. The value zero indicates there is no data following.

MPR\_Type: 16-bits

The MPR\_Type field corresponds to the type of RTP packet. Namely:

0x00: Subflow RTP Header

0x01: Interface Advertisement

0x02: Connectivity Check

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

Flow specific Sequence No.: Sequence of the packet in the subflow. Each subflow has its own strictly monotonically increasing

sequence number space.

MPRTP RTP header extension for Interface Advertisements:

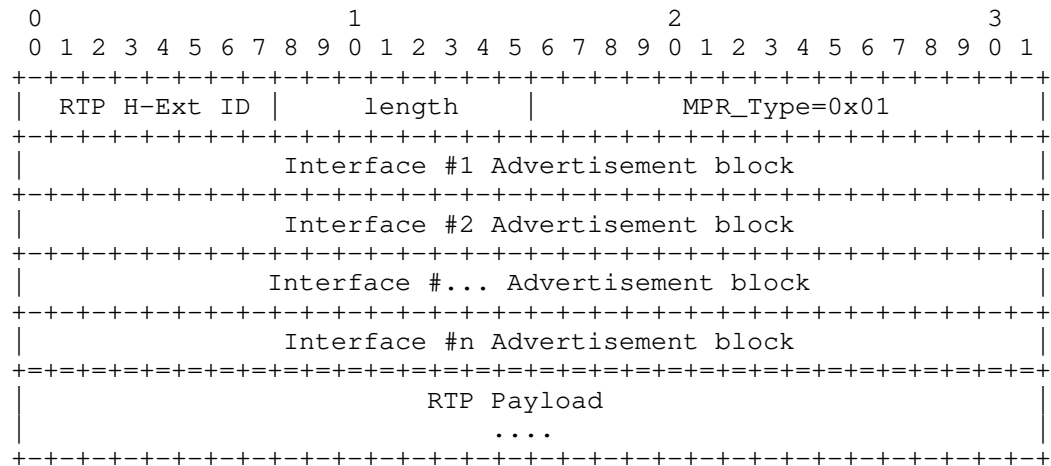


Figure 7: Media Sender's Interface Advertisement (RTP header extension)

Interface Advertisement block: variable size

Defined later in the section.

#### 8.5.2. Interface Address Advertisement block

This 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 RFC 5760 [5].

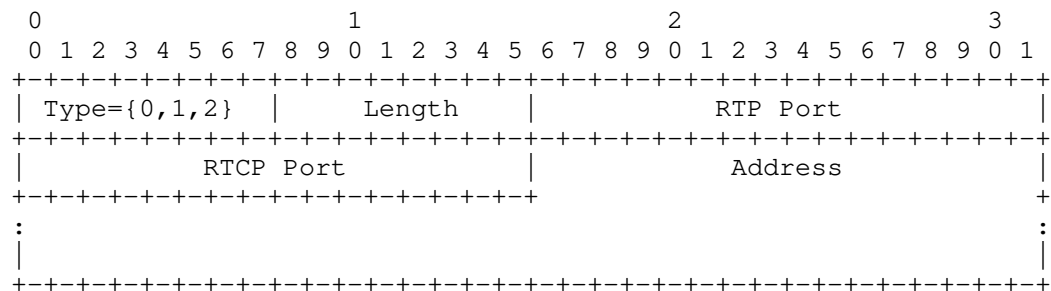


Figure 8: Interface Address Advertisement block during path discovery

Type: 8 bits

The Type corresponds to the type of address. Namely:

0: IPv4 address

1: IPv6 address

2: DNS name

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.

RTCP Port: 2 octets

The port number to which receivers send feedback reports. A port number of 0 is invalid and MUST NOT be used.

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 encoded [6].

#### 8.5.3. MPRTCP RTCP Header Extension

The MPRTCP RTCP header extension is used 1) to provide RTCP feedback per subflow to gauge the characteristics of each path, 2) to advertise the multiple interface addresses for a media receiver, and 3) perform connectivity check on the new interfaces.

MPRTCP RTCP header extension for flow specific SR/RR: TBD

MPRTP RTCP header extension for Interface advertisement:

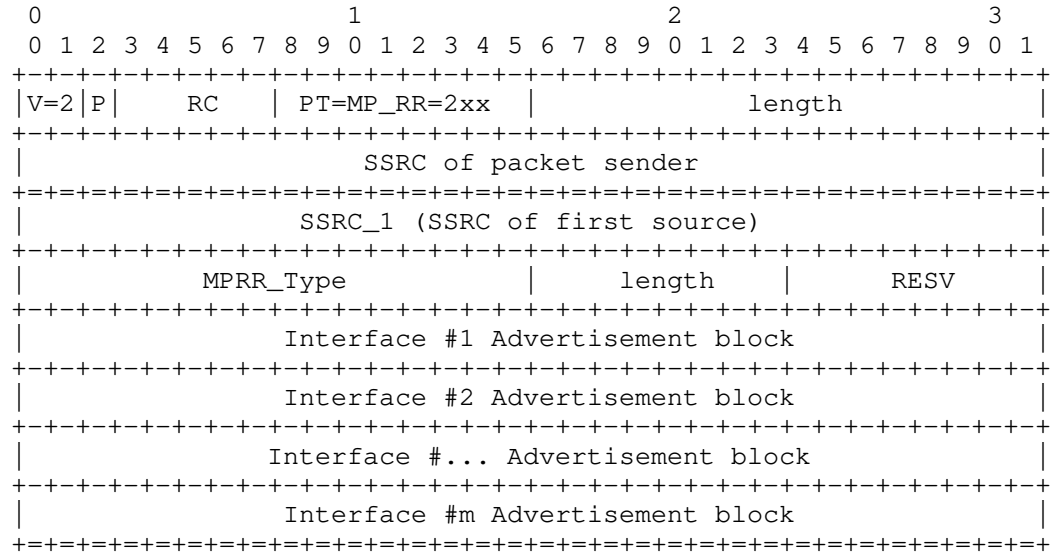


Figure 9: Media Receiver's Interface Advertisement. (RTCP header Extension)

MP\_RR: 8 bits

Indicates that it is a RTCP Receiver Report extension for MPRTP.

MPRR\_Type: 16-bits

The MPRR\_Type field corresponds to the type of MPRTP RTCP packet. Namely:

0x00: Subflow RTCP Statistics Aggregation

0x01: Interface Advertisement

0x02: Connectivity Check

length: 8-bits

The 8-bit length field is the length of extension data in bytes not including the MPRR\_Type and length fields. The value zero indicates there is no data following.

Interface Advertisement block: variable size

Already defined in Section 8.5.2.

## 9. SDP Considerations

The packet formats specified in this document define extensions for RTP and RTCP. The use of MP RTP is left to the discretion of the sender and receiver.

A participant of a media session MAY use SDP to signal that it supports MP RTP. Not providing this information may/will make the sender or receiver ignore the header extensions. However, MP RTP MAY be used by either sender or receiver without prior signaling.

```
mprtp-attr = "a=" "mprtp" [ ":"  
    mprtp-optional-parameter ]  
    CRLF ; flag to enable MP RTP
```

The literal 'mprtp' MUST be used to indicate support for MP RTP. Generally, senders and receivers SHOULD indicate this capability if they support MP RTP and would like to use it in the specific media session being signaled. However, it is possible for an MP RTP sender to stream data using multiple paths to a non-MP RTP client.

Currently, there are no extensions defined for the literal 'mprtp' but we provide the opportunity to extend it using the mprtp-optional-parameter.

### 9.1. 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 peers MUST renegotiate the application specific ("b=AS:") [14] bandwidth.

## 10. 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.

## 11. IANA Considerations

TBD.

## 12. Security Considerations

All drafts are required to have a security considerations section. See RFC 3552 [15] for a guide.

## 13. References

## 13.1. 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] Singer, D. and H. Desineni, "A General Mechanism for RTP Header Extensions", RFC 5285, July 2008.
- [5] Ott, J., Chesterfield, J., and E. Schooler, "RTP Control Protocol (RTCP) Extensions for Single-Source Multicast Sessions with Unicast Feedback", RFC 5760, February 2010.
- [6] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

## 13.2. Informative References

- [7] Stewart, R., "Stream Control Transmission Protocol", RFC 4960, September 2007.
- [8] Ford, A., Raiciu, C., Handley, M., and J. Iyengar, "Architectural Guidelines for Multipath TCP Development", draft-ietf-mptcp-architecture-02 (work in progress), October 2010.
- [9] Nordmark, E. and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6", RFC 5533, June 2009.

- [10] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 5117, January 2008.
- [11] 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.
- [12] Schulzrinne, H., Rao, A., Lanphier, R., Westerlund, M., and M. Stiemerling, "Real Time Streaming Protocol 2.0 (RTSP)", draft-ietf-mmusic-rfc2326bis-25 (work in progress), October 2010.
- [13] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.
- [14] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.
- [15] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.

#### Authors' Addresses

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

Email: varun@comnet.tkk.fi

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  
Nokia Research Center  
P.O. Box 407  
Nokia Group 00045  
Finland

Phone: +358 50 48 24461  
Email: lars.eggert@nokia.com  
URI: [http://research.nokia.com/people/lars\\_eggert](http://research.nokia.com/people/lars_eggert)





Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: April 28, 2011

Q. Wu  
F. Xia  
R. Even  
Huawei  
October 25, 2010

RTCP Report Extension for Feedback Suppression  
draft-wu-avt-retransmission-supression-rtp-06

Abstract

In a large RTP session using the RTCP feedback mechanism defined in RFC 4585, a media source or middlebox may experience transient overload if some event causes a large number of receivers to send feedback at once. This feedback implosion can be mitigated if the device suffering from overload can send a feedback suppression message to the receivers to inhibit further feedback. This memo defines RTCP extensions for feedback suppression, to suppress NACK and FIR feedback requests. It also defines associated SDP signalling."

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 April 28, 2011.

Copyright Notice

Copyright (c) 2010 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.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

## Table of Contents

1. Introduction . . . . .	4
2. Terminology . . . . .	4
3. Protocol Overview . . . . .	5
4. RTCP Feedback Report Extension . . . . .	6
4.1. Transport Layer Feedback: NACK Suppression Report . . . . .	6
4.2. Payload Specific Feedback: FIR suppression report . . . . .	7
5. SDP Signaling . . . . .	8
6. Example Use Cases . . . . .	9
6.1. Source Specific Multicast (SSM) use case . . . . .	9
6.1.1. Simple Feedback Model . . . . .	10
6.1.2. Distribution Source Feedback Summary Model . . . . .	11
6.2. Unicast based Rapid Acquisition of Multicast Stream (RAMS) use case . . . . .	12
6.3. RTP transport translator use case . . . . .	12
6.4. Multipoint Control Unit (MCU) use case . . . . .	13
7. Security Considerations . . . . .	13
8. IANA Consideration . . . . .	14
9. Acknowledgement . . . . .	15
10. References . . . . .	15
10.1. Normative References . . . . .	15
10.2. Informative References . . . . .	16
Authors' Addresses . . . . .	16

## 1. Introduction

RTCP feedback messages [RFC4585] allow the receivers in an RTP session to report events and ask for action from the media source (or a delegated feedback target). There are cases where multiple receivers may initiate the same, or an equivalent message towards the same media source. When the receiver count is large, this behavior may cause transient overload of the media source, the network or both. This is known as a "feedback storm" or a "NACK storm". One common cause of such a feedback storm is receivers utilizing RTP retransmission [RFC4588] as a packet loss recovery technique based, sending feedback using RTCP NACK messages [RFC4585] without proper dithering of the retransmission requests.

Another use case involves video Fast Update requests. A storm of these feedback messages can occur in conversational multimedia scenarios like Topo-Video-switch-MCU [RFC5117]. In this scenario, packet loss may happen on an upstream link of an intermediate network element such as a Multipoint Control Unit (MCU). Poorly designed receivers that blindly issue fast update requests (i.e., Full Intra Request (FIR) described in [RFC5104]), can cause an implosion of FIR requests from receivers to the same media source.

RTCP feedback storms may cause short term overload and, in extreme cases to pose a possible risk of increasing network congestion on the control channel (e.g. RTCP feedback), the data channel, or both. It is therefore desirable to provide a way of suppressing unneeded feedback.

One approach to this, suggested in [DVB-IPTV], involves sending a NACK message from server to the client (or receiver). However NACK is defined as a receiver report sent from a client to the server and therefore exhibits a semantic mismatch when used as a suppression indication from the server (or intermediary) to the client. This document instead specifies a newly message for this function. It further is more precise in the intended uses and less likely to be confusing to receivers. It tells receivers explicitly that feedback for a particular packet or frame loss is not needed and can provide an early indication before the receiver reacts to the loss and invokes its packet loss repair machinery.

## 2. Terminology

The keywords "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. Protocol Overview

This document extends the RTCP feedback messages defined in the Audio-Visual Profile with Feedback (AVPF) and define the Feedback Suppression message. The Feedback Suppression message asks a receiver to not send feedback messages for particular packets (indicated by their RTP sequence numbers) independent of whether the receiver detected the packet loss or detected a need for a decoder refresh point).

In order to detect packet loss before the receivers perceive it, one or more intermediate nodes may be placed between the media source and receiver. These intermediates are variously referred to as Distribution servers, MCUs, RTP translator, or RTP mixers, depending on the precise use case. These intermediaries monitor for packet loss upstream of themselves by checking RTP sequence numbers, just as receivers do. Upon detecting (or suspecting) an upstream loss, the intermediary may send Feedback Suppression message towards the receivers as defined in this specification.

These intermediate nodes need to take into account such factors as the tolerable application delay, the network dynamics, and the media type. When the packet loss is detected upstream of the intermediary and additional latency is tolerable, the intermediate node may itself send a feedback message asking for the suspected lost packet or ask for the correct decoder refresh point. Because it has already provided the necessary feedback toward the source, the intermediate node can be reasonably certain that it will help the situation by sending a Feedback Suppression message to all the relevant receivers, thereby indicating that the receivers should not themselves transmit feedback messages.

Alternatively, the media source may directly monitor the amount of feedback requests it receives, and send feedback suppression messages to the receivers.

When a receiver gets such a feedback suppression message, it should refrain from sending a feedback request (e.g., NACK or FIR) for the missing packets reported in the message. A receiver may still have sent a Feedback message before receiving a feedback suppression message, but further feedback messages for those sequence numbers will be suppressed by this technique. Nodes that do not understand the feedback suppression message will ignore it, and might therefore still send feedback. The media source or intermediate nodes cannot assume that the use of a feedback suppression request actually reduces the amount of feedback it receives.

RTCP Feedback Suppression follows the same semantic model as RTCP

NACK - it conveys the packet receipt/loss events at the sequence number level and considers missing packets as unrepaired. But unlike RTCP NACK, the Feedback Suppression messages can be generated at RTP middleboxes and sent to the corresponding receivers. Intermediaries downstream of an intermediary detecting loss obviously SHOULD NOT initiate their own additional feedback suppression messages for the same packet sequence numbers. They may either simply forward the Feedback Suppression message received from upstream, or augment (or replace) it with a feedback suppression message that reflects the loss pattern they have themselves seen.

Since feedback suppression interacts strongly with repair timing, it has to work together with feedback to not adversely impact the repair of lost source packets. In some cases where the loss was detected and repair initiated much closer to the source, the delay for the receiver to recover from packet loss can be reduced through the combination of intermediary feedback to the source and feedback suppression downstream. In all (properly operating) cases, the risk of increasing network congestion is decreased.

#### 4. RTCP Feedback Report Extension

This document registers two new RTCP Feedback messages for Feedback Suppression. Applications that are employing one or more loss-repair methods MAY use Feedback Suppression together with their existing loss-repair methods either for every packet they expect to receive, or for an application-specific subset of the RTP packets in a session. In other words, receivers MAY ignore Feedback Suppression messages, but SHOULD react to them unless they have good reason to still send feedback messages despite having been requested to suppress them.

##### 4.1. Transport Layer Feedback: NACK Suppression Report

The NACK Implosion Suppression message is an extension to the RTCP feedback report and identified by RTCP packet type value PT=RTPFB and FMT=TBD.

The FCI field MUST contain one or more NACK Suppression Early Indication (NSEI) entries. Each entry applies to a different media source, identified by its SSRC.

The Feedback Control Information (FCI) for NSEI uses the similar format of message Types defined in the section 4.3.1.1 of [RFC5104]. The format is shown in Figure 1.

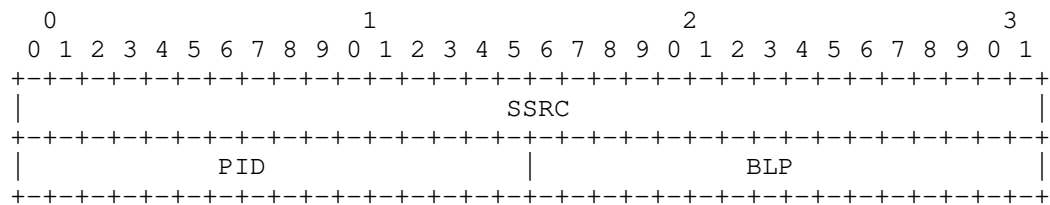


Figure 1: Message Format for the NSEI report

SSRC (32 bits):

The SSRC value of the media source that is requested to send the lost packet.

Packet ID (PID): 16 bits

The PID field is used to specify a lost packet. The PID field refers to the RTP sequence number of the lost packet.

bitmask of proceeding lost packets (BLP): 16 bits

The BLP allows for reporting losses of any of the 16 RTP packets immediately following the RTP packet indicated by the PID. The BLP's definition is identical to that given in [RFC4585].

#### 4.2. Payload Specific Feedback: FIR suppression report

The FIR implosion Suppression message is an extension to the RTCP receiver feedback report and identified by RTCP packet type value PT=PSFB and FMT=TBD.

The FCI field MUST contain one or more FIR suppression Early Indication (FSEI) entries. Each entry applies to a different media source, identified by its SSRC.

The Feedback Control Information (FCI) for FSEI uses the similar format of message Types defined in the section 4.3.1.1 of [RFC5104]. The format is shown in Figure 2.



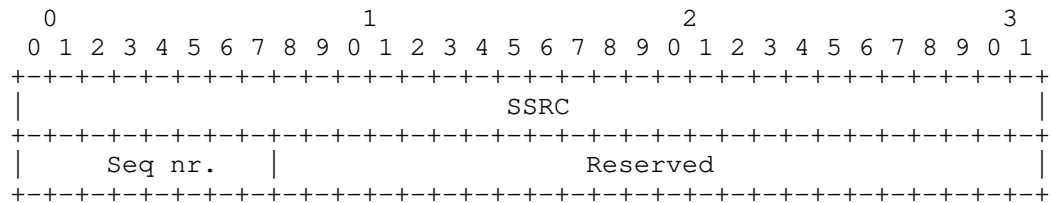


Figure 2: Message Format for the FSEI report

SSRC (32 bits):

The SSRC value of the media source that is requested to send a decoder refresh point.

Seq nr:8bits Command sequence number. The sequence number space is unique for each pairing of the SSRC of command source and the SSRC of the command target. The sequence number SHALL be increased by 1 modulo 256 for each new request.

Reserved: 24 bits

All bits SHALL be set to 0 by the media source and SHALL be ignored on reception.

## 5. SDP Signaling

A new feedback value "fss" needs to be defined for the Feedback Storm Suppression message to be used with Session Description Protocol (SDP) [RFC4566] using the Augmented Backus-Naur Form (ABNF) [RFC4585].

The "fss" feedback value SHOULD be used with parameters that indicate the feedback suppression supported. In this document, we define two such parameters, namely:

- o "fsei" denotes support of fir suppression early indication (fsei).
- o "nsei" denotes support of NACK suppression early indication.

In the ABNF for rtcp-fb-val defined in [RFC4585], there is a placeholder called rtcp-fb-id to define new feedback types. "fss" is defined as a new feedback type in this document, and the ABNF for the parameters for fss is defined here (please refer to section 4.2 of [RFC4585] for complete ABNF syntax).

```
rtcp-fb-val          =/ "fss" rtcp-fb-fss-param
rtcp-fb-fss-param    = SP "nsei";nack suppression early indication
                      / SP "fsei";fir suppression early indication
                      / SP token [SP byte-string]
                      ; for future commands/indications
byte-string = <as defined in section 4.2 of [RFC4585] >
```

Refer to Section 4.2 of [RFC4585] for a detailed description and the full syntax of the "rtcp-fb" attribute.

## 6. Example Use Cases

The operation of feedback suppression is similar for all types of RTP sessions and topologies [RFC5117], however the exact messages used and the scenarios in which suppression is employed differ for various use cases. The following sections outline the intended use cases for feedback suppression and give an overview of the particular mechanisms.

### 6.1. Source Specific Multicast (SSM) use case

In SSM RTP sessions as described in [RFC5760], one or more Media Sources send RTP packets to a Distribution Source. The Distribution Source relays the RTP packets to the receivers using a source-specific multicast group.

In order to avoid the forms of NACK implosion described in section 1, the distribution source should choose to include the support for loss detection. How the packet loss detection works is beyond of scope of this document. When upstream link or downstream aggregate link packet loss occurs, the distribution source creates a Feedback Suppression report and sent it to all the RTP receivers, over the multicast channel. Another possibility is when there may have multiple distribution source placed between the media source and the receivers, the upstream distribution source may inform downstream distribution source of the detected packet loss using Feedback Suppression messages. In response, the distribution source forwards packet loss suppression report received from upstream to all the RTP receivers, over the multicast channel. This loss suppression report tells the receivers that the lost packet will either be forthcoming from distribution source, or it irretrievably lost such that there is nothing to be gained by the receiver sending a NACK to the media source. The distribution source then can (optionally) ask for the lost packets from the media source on behalf of all the RTP receivers.

When there is only one distribution source with loss detection

support between the media source and the receivers, redistribution of the feedback suppression report to all the receivers is trivial. When there are multiple distribution sources between the media source and the receiver, , each distribution source with loss detection support may create a Feedback Suppression Report using the similar format as conventional RTCP NACK packets at the RTP layer and send it to its downstream distribution source or forward one Feedback Suppression Report from upstream to its downstream distribution source or the receivers. Also each distribution source at the downstream of the other distribution source may also create additional Feedback Suppression Report and send it to the receivers.

The distribution source must be able to communicate with all group members in order for either mechanism to be effective at suppressing feedback.

As outlined in the [RFC5760], there are two Unicast Feedback models that may be used for reporting, - the Simple Feedback model and the Distribution Source Feedback Summary Model. The RTCP Feedback Suppression report extension specified in the section 4 of this document will work in both Feedback models. Details of operation in each are specified below.

#### 6.1.1.1. Simple Feedback Model

In the simple Feedback Model, there may have one distribution source with loss detection support. The distribution source must listen on the corresponding RTP session for data. When the distribution source observes that a sequence of RTP packets from upstream contains gaps (by checking the sequence number of packets), the distribution source must use the same packet types as traditional RTCP feedback described in [RFC3550] and create one new RTCP Feedback Report with information on the RTP sequence number of the lost packets and suppression early indication event. When the distribution source is eligible to transmit, it must send this Report packet to the the group on the multicast RTCP session.

Alternatively the distribution source should pass through any feedback suppression requests it receives from the upstream direction. Such a distribution source can also choose to not send feedback suppression messages if it's already seen similar messages with identical packet loss from upstream.

This RTCP Feedback Report lets the receivers know that feedback for this packet loss is not needed and should not be sent to the media source(s). If the media source(s) are part of the SSM group for RTCP packet reflection, the Distribution Source must filter this packet out. If the media source(s) are not part of the SSM group for RTCP

packets, the Distribution Source must not forward this RTCP packets received from the receivers to the media source(s).

#### 6.1.2. Distribution Source Feedback Summary Model

In the distribution source feedback summary model, there may have multiple distribution sources and the Loss Detection instances are distributed into different distribution sources. In some cases, these Loss Detection instances for the same session can exist at the same time, e.g., one Loss Detection instance is implemented in the upstream distribution source A, another two Loss Detection instances for the same session is part of feedback target A and feedback target B respectively within the distribution source B. In this section, we focus on this generic case to discuss the distribution Source Feedback Summary Model.

The distribution source A must listen on the RTP channel for data. When the distribution source A observes RTP packets from a media source are not consecutive by checking the sequence number of packets, the distribution source A generates the new RTCP Feedback Suppression Report packet described in the section 6, and then send it to the distribution source B.

Two loss detection instances within the Distribution Source B must listen for RTCP data sent to the RTCP port. Upon receiving the RTCP Feedback Report packet from the Distribution Source A, the distribution source B needs to summarize the information received from all the RTCP Feedback Reports generated by the upstream distribution source together with the information generated by two loss detection instances within the Distribution Source B and then create the summary report to include all these information. In order to reduce the processing load at the distribution source, each loss detection instance may provide preliminary summarization report.

During the summary report creating, the Distribution Source B must use its own SSRC value for transmitting summarization information and MUST perform proper SSRC collision detection and resolution.

In some case, the distribution source B may receive RTCP NACK messages from the receivers behind the Distribution Source before the distribution source detects the packet loss which may cause potential Feedback implosion. In such case, the distribution source B may filter them out if it already sent a packet loss request for the missing packet to the media source. When the distribution source B confirms packet loss reported by the receiver, the distribution source B generates the summary report to include the packet loss information from the corresponding receiver or upstream distribution source.

The distribution source B may send this new RTCP summary report described in the section 6 to the group on the multicast RTCP channel and in the meanwhile sending a packet loss request to the media source.

If there are a couple of distribution sources with loss detection support looking at the same RTP stream, then the loss may be identified by all and they will all send requests for the same packet loss. In this case, the distribution source must filter out the duplicated information from various distribution source and only append one copy of such information to the summary report.

When the host receives the RTCP Feedback Suppression message, if the host understands this message it will not send packet loss request (e.g., NACK) for the missing packets reported in the message. If it did not understand this new message, the host MAY send packet loss request (e.g., NACK messages) to the specified media source. When the distribution source receives the packet loss request from the hosts after it has already detected packet loss, the distribution source MUST filter it out until proactive recovery is complete.

#### 6.2. Unicast based Rapid Acquisition of Multicast Stream (RAMS) use case

In the typical RAMS architecture, there may have one distribution source placed between media source and BRS for relaying SSM stream from media source. The BRS will receive the SSM stream from the DS. Suppose there are several BRSSes behind the distribution source or media source, there may be just one BRS that detects packet loss on its upstream link between the distribution source and BRS, but the others will perhaps not, as the packet loss took place on SSM tree branch that does not impact the other BRSSes. In such case, the distribution source with loss detection functionality support can not detect packet loss at the downstream of itself, therefore the distribution source SHOULD NOT create new Feedback Suppression message and send it to all the BRS. If BRS impacted by packet loss has loss detection support, the BRS MAY choose to create new Feedback Suppression message and send it to the receivers behind this BRS.

#### 6.3. RTP transport translator use case

A Transport Translator (Topo-Trn-Translator), as defined in [RFC5117] is typically forwarding the RTP and RTCP traffic between RTP clients, for example converting between multicast and unicast for domains that do not support multicast. The translator can identify packet loss from the upstream and send the Feedback Suppression message to the unicast receivers. The translator can also serve as a loss reporter on the multicast side as described in the SSM case.

#### 6.4. Multipoint Control Unit (MCU) use case

In point to multipoint topologies using video switching MCU (Topo-Video-switch-MCU) [RFC5117], the MCU typically forwards a single media stream to each participant, selected from the available input streams. The selection of the input stream is often based on voice activity in the audio-visual conference, but other conference management mechanisms (like presentation mode or explicit floor control) exist as well.

In this case the MCU may detect packet loss from the sender or may decide to switch to a new source. In both cases the receiver may lose synchronization with the video stream and may send a FIR request. If the MCU itself can detect the mis-synchronization of the video, the MCU can send the FIR suppression message to the receivers and send a FIR request to the video source.

#### 7. Security Considerations

The defined messages have certain properties that have security implications. These must be addressed and taken into account by users of this protocol.

Spoofed or maliciously created feedback messages of the type defined in this specification can have the following implications:

Sending NACK Suppression Report with wrong sequence number of lost packet that makes missing RTP packets can not be compensated.

Sending FIR Suppression Report with wrong sequence number of lost packet that makes missing RTP packets can not be compensated by update request mechanism.

To prevent these attacks, there is a need to apply authentication and integrity protection of the feedback messages. This can be accomplished against threats external to the current RTP session using the RTP profile that combines Secure RTP [RFC3711] and AVPF into SAVPF [RFC5124].

Note that middleboxes that are not visible at the RTP layer that wish to send NACK/FIR suppression reports on behalf of the media source can only do so if they spoof the SSRC of the media source. This is difficult in case SRTP is in use. If the middlebox is visible at the RTP layer, this is not an issue, provided the middlebox is part of the security context for the session.

Also note that endpoints that receive a NACK/FIR suppression request

would be well-advised to ignore it, unless it is authenticated via SRTCP or similar. Accepting un-authenticated NACK/ FIR suppression requests can lead to a denial of service attack, where the endpoint accepts poor quality media that could be repaired.

## 8. IANA Consideration

New feedback type and New parameters for RTCP FSS receiver feedback report are subject to IANA registration. For general guidelines on IANA considerations for RTCP feedback, refer to [RFC4585].

This document assigns one new feedback type value x in the RTCP feedback report registry to "Feedback Storm Suppression" with the following registrations format:

Name:	FSS
Long Name:	Feedback Storm Suppression
Value:	TBD
Reference:	This document.

This document also assigns the parameter value y in the RTCP FSS feedback report Registry to "NACK Suppression Early Indication ", with the following registrations format:

Name:	NSEI
Long name:	NACK Suppression Early Indication
Value:	TBD
Reference:	this document.

This document also assigns the parameter value z in the RTCP FSS feedback report Registry to "FIR Suppression Early Indication ", with the following registrations format:

Name:	FSEI
Long name:	FIR Suppression Early Indication
Value:	TBD
Reference:	this document.

The contact information for the registrations is:

Qin Wu  
sunseawq@huawei.com  
101 Software Avenue, Yuhua District  
Nanjing, Jiangsu 210012, China

## 9. Acknowledgement

The authors would like to thank David R Oran, Ali C. Begen, Colin Perkins, Tom VAN CAENEGEM, Ingemar Johansson S, Bill Ver Steeg, WeeSan Lee for their valuable comments and suggestions on this document.

## 10. References

### 10.1. Normative References

- [RFC5760] Ott, J., Chesterfield, J., and E. Schooler, "RTP Control Protocol (RTCP) Extensions for Single-Source Multicast Sessions with Unicast Feedback", RFC 5760, February 2010.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4585] 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.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [RFC5117] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 5117, January 2008.
- [RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R. Hakenberg, "RTP Retransmission Payload Format", RFC 4588, July 2006.
- [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.
- [RFC5104] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, February 2008.
- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.



- [RFC5124] Ott, J. and E. Carrara, "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)", RFC 5124, February 2008.

## 10.2. Informative References

- [RFC5740] Adamson, B., Bormann, C., Handley, M., and J. Macker, "NACK-Oriented Reliable Multicast (NORM) Transport Protocol", November 2009.
- [DVB-IPTV]  
ETSI Standard, "Digital Video Broadcasting (DVB); Transport of MPEG-2 TS Based DVB Services over IP Based Networks", ETSI TS 102 034, V1.4.1, August 2009.
- [I-D.hunt-avt-monarch-01]  
Hunt, G. and P. Arden, "Monitoring Architectures for RTP", August 2008.
- [I-D.ietf-pmol-metrics-framework-02]  
Clark, A., "Framework for Performance Metric Development".

## Authors' Addresses

Qin Wu  
Huawei  
101 Software Avenue, Yuhua District  
Nanjing, Jiangsu 210012  
China

Email: sunseawq@huawei.com

Frank Xia  
Huawei  
1700 Alma Dr. Suite 500  
Plano, TX 75075  
USA

Phone: +1 972-509-5599  
Email: xiayangsong@huawei.com

Roni Even  
Huawei  
14 David Hamelech  
Tel Aviv 64953  
Israel

Email: [even.roni@huawei.com](mailto:even.roni@huawei.com)



Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: April 16, 2011

Q. Wu  
Huawei  
G. Zorn  
Network Zen  
R. Schott  
Deutsche Telekom Laboratories  
October 13, 2010

RTP Control Protocol Extended Reports (RTCP XR) Report Blocks for Real-  
time Video Quality Monitoring  
draft-wu-avt-rtcp-xr-quality-monitoring-04

Abstract

This document defines a set of RTP Control Protocol Extended Reports (RTCP XR) Report Blocks and associated SDP parameters allowing the report of video quality metrics, primarily for video applications of RTP.

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 April 16, 2011.

Copyright Notice

Copyright (c) 2010 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 . . . . .	3
2. Terminology . . . . .	4
2.1. Standards Language . . . . .	4
2.2. Acronyms . . . . .	5
3. Applicability . . . . .	5
4. Transport Layer Metrics . . . . .	5
4.1. RTP Flows Initial Synchronization Delay Report Block . . . . .	6
4.2. RTP Flow General Synchronization Offset Metrics Block . . . . .	7
4.3. Layered Streams Statistics Metrics Block . . . . .	8
5. Application Layer Metrics . . . . .	9
5.1. RTP Streams Statistics Summary Report Block . . . . .	9
5.2. Video Stream Loss and Discard Metrics Block . . . . .	11
5.3. Video Stream Burst Metrics Block . . . . .	13
5.4. Synthetical Multimedia Quality Metrics Block . . . . .	15
6. SDP Signaling . . . . .	17
7. IANA Considerations . . . . .	19
8. Security Considerations . . . . .	20
9. Acknowledgements . . . . .	20
10. References . . . . .	20
10.1. Normative References . . . . .	20
10.2. Informative References . . . . .	21
Authors' Addresses . . . . .	22

## 1. Introduction

Along with the wide deployment of broadband access and the development of new IPTV services (e.g., broadcast video, video on demand), there is increasing interest in monitoring and managing networks and applications that deliver real-time applications over IP, to ensure that all end users obtain acceptable video/audio quality. The main drives come from operators, since offering performance monitoring capability can help diagnose network impairments, facilitate in root cause analysis and aid in verifying compliance with service level agreements (SLAs) between Internet Service Providers (ISPs) and content providers.

The factors that affect real-time application quality can be split into two categories. The first category consists of transport-dependent factors such as packet loss, delay and jitter (which also translates into losses in the playback buffer). The factors in the second category are application-specific factors that affect video quality and are sensitivity to network errors. These factors can be but not limited to video codec and loss recovery technique, coding bit rate, packetization scheme, and content characteristics.

Compared with application-specific factors, the transport-dependent factors sometimes are not sufficient to measure video quality, since the ability to analyze the video in the application layer provides quantifiable measurements for subscriber Quality of Experience (QoE) that may not be captured in the transmission layers or from the RTP layer down. In a typical scenario, monitoring of the transmission layers can produce statistics suggesting that quality is not an issue, such as the fact that network jitter is not excessive. However, problems may occur in the service layers leading to poor subscriber QoE. Therefore monitoring using only network-level measurements may be insufficient when application layer video quality is required.

In order to provide accurate measures of video quality for operators when transporting video across a network, the video quality Metrics is highly required which can be conveyed in the RTCP XR packets[RFC3611] and may have the following three benefits:

- \* Tuning the video encoder algorithm to satisfy video quality requirements
- \* Determining which system techniques to use in a given situation and when to switch from one technique to another as system parameters change
- \* Verifying the continued correct operation of an existing system

RFC 3611 [RFC3611] defines seven report block formats for network

management and quality monitoring. However, there are no block types specifically designed for conveying video quality metrics. This document focuses on specifying new report block types used to convey video-specific quality metrics.

The report block types defined in this document fall into two categories. The first category consists of general information regarding transmission quality, to be generated and processed by the RTP transport. The report blocks in the second category convey metrics above transport that affect video quality and are sensitivity to network errors.

Seven report block formats are defined by this document. Of these, three are transport layer metrics:

- \* RTP Flows Initial Synchronization Delay Report Block
- \* Audio-Video Playout Offset Report Block
- \* Layered Streams Statistics Metrics Block

The other four are application layer metrics:

- \* Video Statistics Summary Report Block
- \* Video Stream Loss and Discard Metrics Block
- \* Video Stream Burst Metrics Block
- \* Synthetical Multimedia Quality Metrics Block

## 2. Terminology

### 2.1. Standards 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 RFC 2119 [RFC2119].

In addition, the following terms are defined:

Layered Component Packet

a RTP packet using layered codecs containing the specified layered component, e.g., encoded stream at the base layer or at the enhancement layer.

Picture Type

Picture types used in the different video algorithms compose of the key-frame and the Derivation frame. Key-frame is also called as reference frame and used as a reference for predicting other

pictures. It is coded without prediction from other pictures. The Derivation frame is derived from Key-frame using prediction from the reference frame.

## 2.2. Acronyms

SSRC

Synchronization Source [RFC3550]

TS

Transport Stream [ISO-IEC.13818-1.2007]

## 3. Applicability

All the report blocks defined in this document could be used by dedicated network monitoring applications. As specified in RFC 3611 [RFC3611], for such an application it might be appropriate to allow more than 5% of RTP data bandwidth to be used for RTCP packets, thus allowing proportionately larger and more detailed report blocks.

The Flows General Synchronization Offset Block Section 4.2 has been defined for various multimedia applications. Such applications can use this report block to monitor offset between two RTP streams synchronization to ensure satisfactory QoE. Tighter tolerances than typically used have been recommended for such applications.

The Flows Synchronization Delay Report Block has been defined primarily for layered or multi-description video coding applications. When joining a layered video session in such an application, a receiver may not synchronize playout across the multimedia session until RTCP SR packets have been received on all of the component RTP sessions. This report block can be used to ensure synchronization between different media layers for the same multimedia session.

The Video Stream Loss and Discard Metrics Report Block, Video Stream Burst Metrics report Block, Video Statistics Summary Report Block and Layered Video Statistics Metrics Block can be applied to any real time video application, while Synthetical Multimedia Quality Metrics Report Block can be used in any real-time AV application .

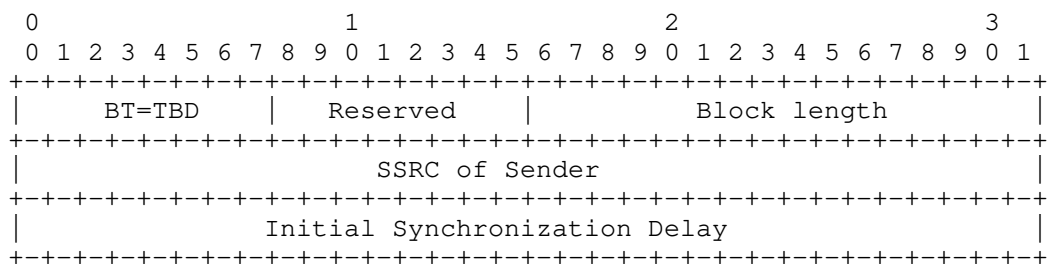
## 4. Transport Layer Metrics



#### 4.1. RTP Flows Initial Synchronization Delay Report Block

This block reports the initial synchronization delay between RTP sessions of the same media stream sent using Multi-Session Transmission [I-D.ietf-avt-rtp-svc] or the initial synchronization delay between RTP session of the different media types [I-D.ietf-avt-rapid-rtp-sync], which is beyond the information carried in the standard RTCP packet format. Information is recorded about session bandwidth and synchronization delay.

The RTP Flows Initial Synchronization Delay Report Block has the following format:



Block type (BT): 8 bits

The Statistics Summary Report Block is identified by the constant <RFISD>.

Reserved: 8 bits

This field is reserved for future definition. In the absence of such a definition, the bits in this field MUST be set to zero and MUST be ignored by the receiver.

Block length: 16 bits

The constant 3, in accordance with the definition of this field in Section 3 of RFC 3611 [RFC3611].

SSRC of Sender: 32 bits

The SSRC of the RTP data packet source being reported upon by this report block. (Section 4.1 of [RFC3611]).

Initial Synchronization Delay: 32 bits

The average delay, expressed in units of 1/65536 seconds, between the RTCP packets received on all of the components RTP sessions and the beginning of session [I-D.ietf-avt-rapid-rtp-sync]. The value is calculated as follows:

The average time, expressed in units of 1/65536 seconds, taken to receive the first RTCP packet in the RTP session with the longest RTCP reporting interval [I-D.ietf-avt-rapid-rtp-sync]

#### 4.2. RTP Flow General Synchronization Offset Metrics Block

In an RTP multimedia session, there can be an arbitrary number of streams, with the same RTCP CNAME. This block reports the general Synchronization offset requirements of these RTP streams beyond the information carried in the standard RTCP packet format. Information is recorded about the synchronization offset time of each RTP stream relative to the reference RTP stream with the same CNAME and General Synchronisation Offset of zero.. The RTP Flow General Synchronization Offset Report Block has the following format:

```

      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
+-----+-----+-----+-----+-----+-----+-----+-----+
|      BT=TBD      | I |  Reserved  |          Block length          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     SSRC of source                 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     General Synchronization Offset   |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Block type (BT): 8 bits

The Statistics Summary Report Block is identified by the constant <AVPO>.

Interval Metric flag (I): 1 bit

This field is used to indicate whether the Audio-Video synchronization metrics are Interval or Cumulative metrics, that is, whether the reported values applies to the most recent measurement interval duration between successive metrics reports (I=1) (the Interval Duration) or to the accumulation period characteristic of cumulative measurements (I=0) (the Cumulative Duration).

Reserved: 8 bits

This field is reserved for future definition. In the absence of such a definition, the bits in this field MUST be set to zero and MUST be ignored by the receiver.

Block length: 16 bits

The constant 2, in accordance with the definition of this field in Section 3 of RFC 3611 [RFC3611].

SSRC of source: 32 bits

As defined in Section 4.1 of RFC 3611 [RFC3611].

General synchronization offset: 32 bits

This field indicates the synchronization offset time of one RTP stream in milliseconds relative to the reference RTP stream with the same CNAME and General Synchronisation Offset of zero [I-D.ietf-avt-rapid-rtcp-sync] This value is calculated based on the interarrival time between arbitray RTP packet and the reference RTP packet with the same CNAME , and timestamps of this arbitray RTP packet and the reference RTP packet with the same CNAME.

#### 4.3. Layered Streams Statistics Metrics Block

This block reports layered streams statistics beyond the information carried in the Statistics Summary Report Block RTCP packet specified in the section 4.6 of RFC 3611 [RFC3611]. Information is recorded about lost layered component packets, duplicated layered component packets. Such information can be useful for network management and video quality monitoring.

The report block contents are dependent upon a series of flag bits carried in the first part of the header. Not all parameters need to be reported in each block. Flags indicate which parameters are reported and which are not. The fields corresponding to unreported parameters MUST be present, but are set to zero. The receiver MUST ignore any Layered Streams Statistics Metrics Block with a non-zero value in any field flagged as unreported.

The Layered Stream Statistics metrics Block has the following format:

```

      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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      BT=TBD      |T|      rsd.      |      block length      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     SSRC of source                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      begin_seq      |      end_seq      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Lost_Layered Component Packets                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Dup Layered Component_Packets                                     |

```

[illegible]

Block type (BT): 8 bits

The Layered stream Statistics Metrics Block is identified by the constant <LSSM>.

Layer Type flag (T): 1 bits

This field is used to indicate the Layer Type of layered video to be reported. LT is set to 0 if the loss\_component\_packet field and dup\_component packet contain the base layer packet in layered codecs, e.g, SVC in [I-D.ietf-avt-rtp-svc], 1 if the loss\_component packet field and dup\_component packet contain enhancement layer packet in layered codec.

Rsd.: 3 bits

This field is reserved for future definition. In the absence of such a definition, the bits in this field MUST be set to zero and MUST be ignored by the receiver.

Block length: 16 bits

The constant 3, in accordance with the definition of this field in Section 3 of RFC 3611 [RFC3611].

SSRC of source: 32 bits

As defined in Section 4.1 of RFC 3611 [RFC3611].

```
begin_seq: 16 bits
```

As defined in Section 4.1 of RFC 3611 [RFC3611].

end\_seq: 16 bits

As defined in Section 4.1 of RFC 3611 [RFC3611].

Lost\_Layered Component Packets: 32 bits

Number of lost\_component packets in the above sequence number interval.

Dup\_Layered Component Packets: 32 bits

Number of dup\_component packets in the above sequence number interval.

## 5. Application Layer Metrics

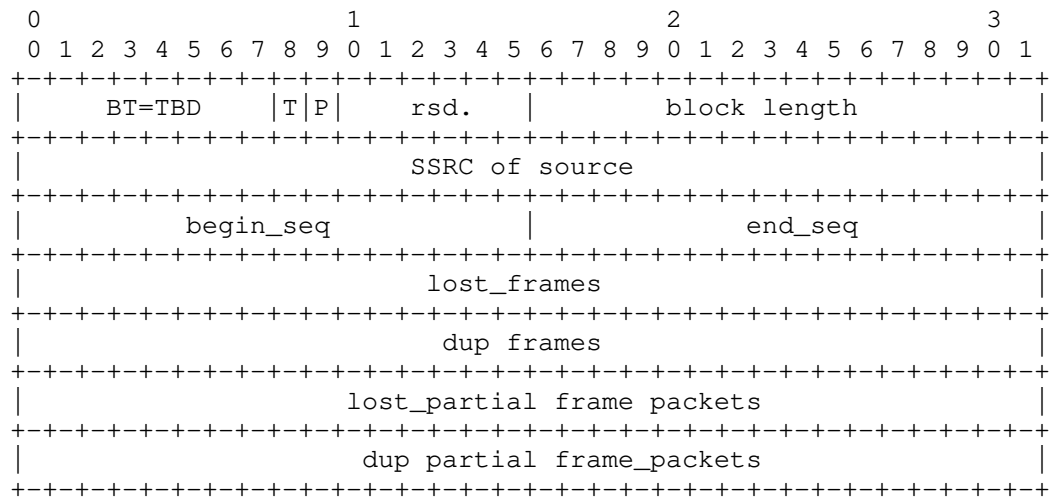
### 5.1. RTP Streams Statistics Summary Report Block

This block reports statistics beyond the information carried in the Statistics Summary Report Block RTPC packet specified in the section

4.6 of RFC 3611 [RFC3611]. Information is recorded about lost frame packets, duplicated frame packets, lost layered component packets, duplicated layered component packets. Such information can be useful for network management and video quality monitoring.

The report block contents are dependent upon a series of flag bits carried in the first part of the header. Not all parameters need to be reported in each block. Flags indicate which parameters are reported and which are not. The fields corresponding to unreported parameters MUST be present, but are set to zero. The receiver MUST ignore any Video Statistics Summary Report Block with a non-zero value in any field flagged as unreported.

The RTP Streams Statistics Summary Report Block has the following format:



Block type (BT): 8 bits

The Video Statistics Summary Report Block is identified by the constant <VSS>.

Picture type indicator (T): 1 bits

Picture types used in the different video algorithms compose of key-frame and derivation frame. This field is used to indicate the frame type to be reported. Bits set to 0 if the lost\_frames field or dup\_frames field contain a key\_frame report or reference frame report, 1 if the lost\_frames field and dup\_frames field contain other derivation frame report.

P: 1 bit

Bit set to 1 if the `lost_partial` frame packets field or the `dup_partial_frame` packets field contains a report, 0 otherwise.

Rsd.: 3 bits

This field is reserved for future definition. In the absence of such a definition, the bits in this field **MUST** be set to zero and **MUST** be ignored by the receiver.

Block length: 16 bits

The constant 5, in accordance with the definition of this field in Section 3 of RFC 3611 [RFC3611].

SSRC of source: 32 bits

As defined in Section 4.1 of RFC 3611 [RFC3611].

begin\_seq: 16 bits

As defined in Section 4.1 of RFC 3611 [RFC3611].

end\_seq: 16 bits

As defined in Section 4.1 of RFC 3611 [RFC3611].

lost\_frames: 32 bits

Number of `lost_frames` in the above sequence number interval.

dup\_frames: 32 bits

Number of `dup_frames` in the above sequence number interval.

lost\_partial frame packets: 32 bits

Number of `lost_partial` frame packets in the above sequence number interval.

dup\_partial frame packets: 32 bits

Number of `dup_partial` frame packets in the above sequence number interval.

## 5.2. Video Stream Loss and Discard Metrics Block

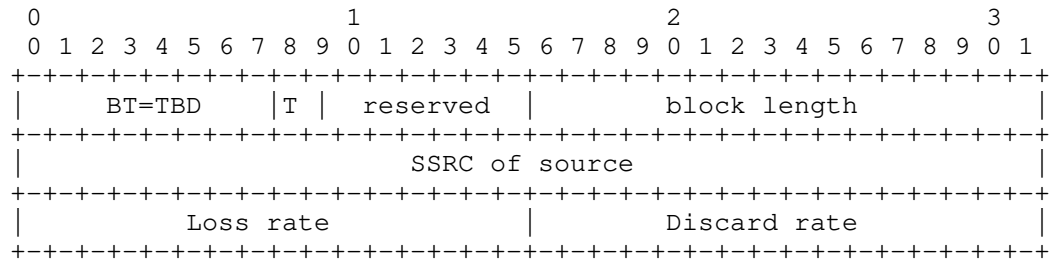
This block reports Loss and Discard metrics statistics beyond the information carried in the standard RTCP packet format. The block reports separately on packets lost on the IP channel, and those that have been received but then discarded by the receiving jitter buffer.

It is very useful to distinguish between packets lost by the network and those discarded due to jitter. Both have equal effect on the quality of the video stream, however, having separate counts helps identify the source of quality degradation. These fields **MUST** be

populated, and MUST be set to zero if no packets have been received.

Implementations MUST provide values for all the fields defined here. For certain metrics, if the value is undefined or unknown, then the specified default or unknown field value MUST be provided.

The block is encoded as six 32-bit words:



block type (BT): 8 bits

A Video Stream Metrics Report Block is identified by the constant <VSLDM>.

Picture type indicator (T): 1 bits

Picture types used in the different video algorithms compose of key-frame and derivation frame. This field is used to indicate the picture type to be reported. Bits set to 0 if the Loss rate field and discard rate field contain a Key\_frame report or reference frame report, 1 if the Loss rate field and discard rate field contain other derivation frame reports.

reserved: 6 bits

This field is reserved for future definition. In the absence of such a definition, the bits in this field MUST be set to zero and MUST be ignored by the receiver.

block length: 16 bits

The constant 1, in accordance with the definition of this field in Section 3 of RFC 3611 [RFC3611].

SSRC of source: 32 bits

The SSRC of the RTP data packet source being reported upon by this report block. in accordance with the definition of this field in Section 3 of RFC 3611 [RFC3611].

Loss rate: 8 bits

The fraction of RTP data packets from the source lost since the beginning of reception, expressed as a fixed point number with the binary point at the left edge of the field. This value is calculated by dividing the total number of lost packets containing

specified frame (e.g., Key frame) (after the effects of applying any error protection such as FEC) by the total number of packets expected, multiplying the result of the division by 256, limiting the maximum value to 255 (to avoid overflow), and taking the integer part. The numbers of duplicated packets and discarded packets do not enter into this calculation. Since receivers cannot be required to maintain unlimited buffers, a receiver MAY categorize late-arriving packets as lost. The degree of lateness that triggers a loss SHOULD be significantly greater than that which triggers a discard.

Discard rate: 8 bits

The fraction of RTP data packets from the source that have been discarded since the beginning of reception, due to late or early arrival, under-run or overflow at the receiving jitter buffer. This value is expressed as a fixed point number with the binary point at the left edge of the field. It is calculated by dividing the total number of discarded packets containing specified frame (e.g., Key Frame) (excluding duplicate packet discards) by the total number of packets expected, multiplying the result of the division by 256, limiting the maximum value to 255 (to avoid overflow), and taking the integer part.

### 5.3. Video Stream Burst Metrics Block

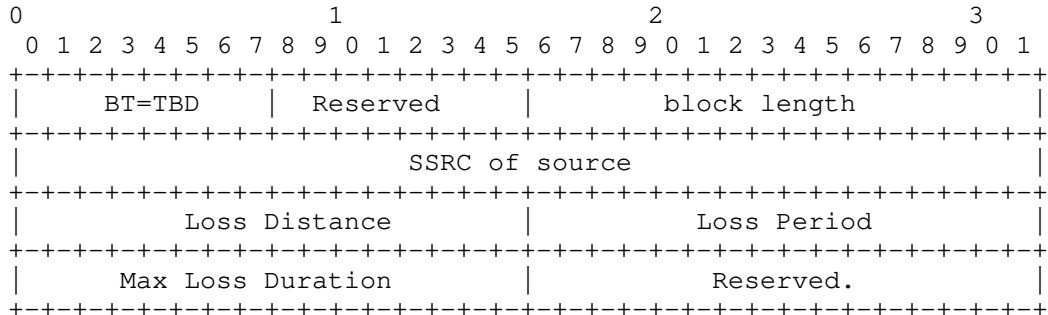
This block reports Burst metrics statistics beyond the information carried in the standard RTCP packet format. It reports on the combined effect of losses and discards, as both have equal effect on video quality.

In order to properly assess the quality of a video stream, it is desirable to consider the degree of burstiness of packet loss RFC 3357 [RFC3357]. Following the one-way loss pattern sample metrics discussed in [RFC3357], a measure of the spacing between consecutive network packet loss or error events, is a "loss distance". The loss distance metric captures the spacing between the loss periods. The duration of a loss or error event (e.g. and how many packets are lost in that duration) is a "loss period", the loss period metric captures the frequency and length (burstiness) of loss once it starts. Delay reports include the transit delay between RTP end points and the end system processing delays, both of which contribute to the user perceived delay.

Implementations MUST provide values for all the fields defined here. For certain metrics, if the value is undefined or unknown, then the specified default or unknown field value MUST be provided.



The block is encoded as six 32-bit words:



block type (BT): 8 bits

A Video Stream Metrics Report Block is identified by the constant <VSBM>.

reserved: 8 bits

This field is reserved for future definition. In the absence of such a definition, the bits in this field MUST be set to zero and MUST be ignored by the receiver.

block length: 16 bits

The constant 2, in accordance with the definition of this field in Section 3 of RFC 3611 [RFC3611].

SSRC of source: 32 bits

The SSRC of the RTP data packet source being reported upon by this report block. in accordance with the definition of this field in Section 3 of RFC 3611 [RFC3611].

Loss Distance: 16 bits

The mean duration, expressed in milliseconds, of the loss intervals that have occurred since the beginning of reception [DSLRF]. The duration of each loss distance is calculated based upon the frames that mark the beginning and end of that period. It is equal to the timestamp of the end frame, plus the duration of the end frame, minus the timestamp of the beginning frame. If the actual values are not available, estimated values MUST be used. If there have been no burst periods, the burst duration value MUST be zero.

Loss Period: 16 bits

The mean duration, expressed in milliseconds, of the burst loss periods that have occurred since the beginning of reception [DSLRF]. The duration of each period is calculated based upon the frame that marks the end of the prior burst loss and the frame

that marks the beginning of the subsequent burst loss. It is equal to the timestamp of the subsequent burst frame, minus the timestamp of the prior burst packet, plus the duration of the prior burst packet. If the actual values are not available, estimated values MUST be used. In the case of a gap that occurs at the beginning of reception, the sum of the timestamp of the prior burst packet and the duration of the prior burst packet are replaced by the reception start time. In the case of a gap that occurs at the end of reception, the timestamp of the subsequent burst packet is replaced by the reception end time. If there have been no gap periods, the gap duration value MUST be zero.

Max Loss Duration of a single error: 16 bits

The maximum loss duration, expressed in milliseconds, of the loss periods that have occurred since the beginning of reception. The recommended max loss duration is specified as less than 16 ms in [DSLIF], which provides a balance between interleaver depth protection from xDSL errors induced by impulse noise, delay added to other applications and video service QoE requirements to reduce visible impairments.

Reserved: 16 bits

All bits SHALL be set to 0 by the sender and SHALL be ignored on reception.

block length: 16 bits

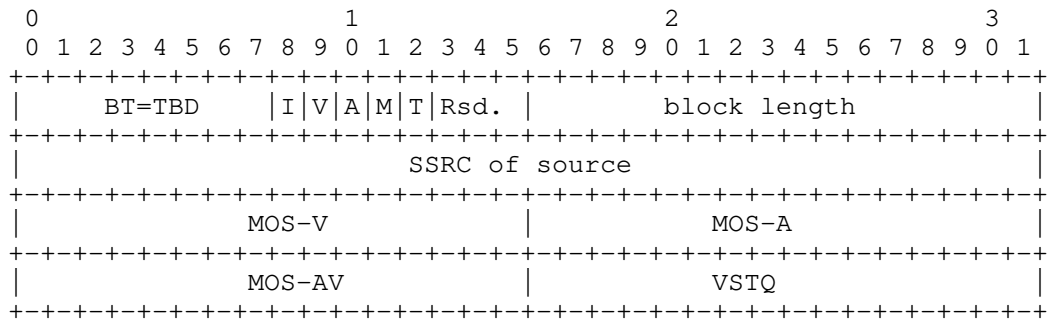
The constant 2, in accordance with the definition of this field in Section 3 of RFC 3611 [RFC3611].

#### 5.4. Synthetical Multimedia Quality Metrics Block

This block reports the multimedia quality metrics beyond the information carried in the standard RTCP packet format. Information is recorded about Video MOS, Audio MOS, Audio Video MOS, Video Service Transmission Quality [G.1082][P.NAMS].

The report block contents are dependent upon a series of flag bits carried in the first part of the header. Not all parameters need to be reported in each block. Flags indicate which are and which are not reported. The fields corresponding to unreported parameters MUST be present, but are set to zero. The receiver MUST ignore any Perceptual Quality Metrics Block with a non-zero value in any field flagged as unreported.

The Synthetical Multimedia Quality Metrics Block has the following format:



Block type (BT): 8 bits

The Perceptual Quality Metrics Block is identified by the constant <SMQM>.

Interval Metric flag (I): 1 bit

This field is used to indicate whether the Basic Loss/Discard metrics are Interval or Cumulative metrics, that is, whether the reported values applies to the most recent measurement interval duration between successive metrics reports (I=1) (the Interval Duration) or to the accumulation period characteristic of cumulative measurements (I=0) (the Cumulative Duration).

MOS-V flag (V): 1 bit

Bit set to 1 if the MOS-V field and MOS-AV field contain a report, 0 otherwise.

MOS-A flag (A): 1 bit

Bit set to 1 if the MOS-A field contain a report, 0 otherwise.

MOS-AV flag (M): 1 bit

Bit set to 1 if the MOS-AV field contain a report, 0 otherwise.

Video Service Transmission Quality flag (T): 1 bit

Bit set to 1 if the VSTQ field contains a report, 0 otherwise.

Rsd.: 3 bits

This field is reserved for future definition. In the absence of such a definition, the bits in this field MUST be set to zero and MUST be ignored by the receiver.

SSRC of source: 32 bits

As defined in Section 4.1 of [RFC3611].

**MOS-V: 16 bits**

The estimated mean opinion score for video quality (MOS-V) is a video quality metric on a scale from 1 to 5, in which 5 represents excellent and 1 represents unacceptable [G.1082][P.NAMS]. This metric is defined as not including the effects of audio impairments and can be compared to MOS scores obtained from video quality tests. It is expressed as an integer in the range 10 to 50, corresponding to MOS x 10. For example, a value of 35 would correspond to an estimated MOS score of 3.5.

A value of 127 indicates that this parameter is unavailable. Values other than 127 and the valid range defined above MUST NOT be sent and MUST be ignored by the receiving system.

**MOS-A: 16 bits**

The estimated mean opinion score for Audio quality (MOS-A) is defined as including the effects of delay and other effects that would affect Audio-Video quality [RFC3611]. It is expressed as an integer in the range 10 to 50, corresponding to MOS x 10, as for MOS-A.

A value of 127 indicates that this parameter is unavailable. Values other than 127 and the valid range defined above MUST not be sent and MUST be ignored by the receiving system.

**MOS-AV: 16 bits**

The estimated mean opinion score for Audio-Video quality (MOS-AV) is defined as including the effects of delay and other effects that would affect Audio-Video quality [G.1082][P.NAMS]. It is expressed as an integer in the range 10 to 50, corresponding to MOS x 10, as for MOS-AV. A value of 127 indicates that this parameter is unavailable. Values other than 127 and the valid range defined above MUST NOT be sent and MUST be ignored by the receiving system.

**VSTQ: 16 bits**

Video Service Transmission Quality (TBC) .

## 6. SDP Signaling

Six new parameters are defined for the six report blocks defined in this document to be used with Session Description Protocol (SDP) [RFC4566] using the Augmented Backus-Naur Form (ABNF) [RFC5234]. They have the following syntax within the "rtcp-xr" attribute [RFC3611]:

```
rtcp-xr-attr = "a=rtcp-xr:"  
               [xr-format *(SP xr-format)] CRLF  
  
xr-format = RTP-flows-syn  
           / audio-video-offset  
           / multimedia-quality-metrics  
           / video-stream-loss-metrics  
           / video-stream-burst-metrics  
           / video-stat-summary  
           / layered-video-stat-metrics  
  
RTP-flows-syn = "RTP-flows-syn"  
               ["=" max-size]  
               max-size = 1*DIGIT ; maximum block size in octets  
  
audio-video-offset = "audio-video-offset"  
                   ["=" max-size]  
                   max-size = 1*DIGIT ; maximum block size in octets  
  
video-stream-burst-metrics = "video-stream-burst-metrics"  
                           ["=" max-size]  
                           max-size = 1*DIGIT ; maximum block size in octets  
  
video-stream-loss-metrics = "video-stream-loss-metrics"  
                           ["=" stat-flag *(", " stat-flag)]  
                           stat-flag = "key Frame loss and duplication"  
                                       / "derivation Frame loss and duplication"  
  
video-stat-summary = "video-stat-summary"  
                    ["=" stat-flag *(", " stat-flag)]  
                    stat-flag = "key Frame loss and duplication"  
                                / "derivation Frame loss and duplication"  
  
layered-stream-stat-metrics = "layered-stream-stat-metrics"  
                             ["=" stat-flag *(", " stat-flag)]  
                             stat-flag = "base layer packet"  
                                         / "enhancement layer packet"  
  
multimedia-quality-metrics = "multimedia-quality-metrics"  
                             ["=" stat-flag *(", " stat-flag)]  
                             stat-flag = "Interval Metric"  
                                         / "MOS-V"  
                                         / "MOS-A"  
                                         / "MOS-AV"  
                                         / "VSTQ"
```

Refer to Section 5.1 of RFC 3611 [RFC3611] for a detailed description

and the full syntax of the "rtcp-xr" attribute.

## 7. IANA Considerations

New report block types for RTCP XR are subject to IANA registration. For general guidelines on IANA allocations for RTCP XR, refer to Section 6.2 of [RFC3611].

This document assigns six new block type values in the RTCP XR Block Type Registry:

Name: RFISD  
Long Name: RTP Flows Initial Synchronization Delay  
Value: <RFISD>  
Reference: Section 4.1

Name: AVPO  
Long Name: Audio-Video Playout Offset  
Value: <AVPO>  
Reference: Section 4.2

Name: VSS  
Long Name: Video Statistics Summary  
Value: <VSS>  
Reference: Section 5.1

Name: LSSM  
Value: <LSSM>  
Long Name: Layered Stream Statistics Metrics  
Reference: Section 4.3

Name: VSLDM  
Long Name: Video Stream Loss and Discard Metrics  
Value: <VSLDM>  
Reference: Section 5.2

Name: VSBM  
Long Name: Video Stream Burst Metrics  
Value: <VSBM>  
Reference: Section 5.3

Name: SMQM  
Long Name: Synthetical Multimedia Quality Metric

Value            <SMQM>  
Reference:    Section 5.4

This document also registers seven SDP [RFC4566] parameters for the "rtcp-xr" attribute in the RTCP XR SDP Parameters Registry:

- \* "RTP-flows-syn"
- \* "audio-video-offset"
- \* "multimedia-quality-metrics"
- \* "video-stream-loss-metrics"
- \* "video-stream-burst-metrics"
- \* "video-stat-summary"
- \* "layered-stream-stat-metrics"

The contact information for the registrations is:

Qin Wu  
sunseawq@huawei.com  
101 Software Avenue, Yuhua District  
Nanjing, JiangSu 210012 China

## 8. Security Considerations

TBC

## 9. Acknowledgements

The authors would like to thank Bill Ver Steeg, David R Oran, Ali Begen, Colin Perkins, Roni Even, Youqing Yang, Wenxiao Yu and Yinliang Hu for their valuable comments and suggestions on this document.

## 10. References

### 10.1. Normative References

- [I-D.ietf-avt-rapid-rtp-sync]  
Perkins, C. and T. Schierl, "Rapid Synchronisation of RTP Flows", draft-ietf-avt-rapid-rtp-sync-12 (work in progress), July 2010.
- [I-D.ietf-avt-rtp-svc]  
Wenger, S., Wang, Y., Schierl, T., and A. Eleftheriadis, "RTP Payload Format for Scalable Video Coding", draft-ietf-avt-rtp-svc-23 (work in progress), October 2010.

- [ISO-IEC.13818-1.2007]  
International Organization for Standardization,  
"Information technology - Generic coding of moving  
pictures and associated audio information: Systems",  
ISO International Standard 13818-1, October 2007.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate  
Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2250] Hoffman, D., Fernando, G., Goyal, V., and M. Civanlar,  
"RTP Payload Format for MPEG1/MPEG2 Video", RFC 2250,  
January 1998.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V.  
Jacobson, "RTP: A Transport Protocol for Real-Time  
Applications", STD 64, RFC 3550, July 2003.
- [RFC3611] Friedman, T., Caceres, R., and A. Clark, "RTP Control  
Protocol Extended Reports (RTCP XR)", RFC 3611,  
November 2003.
- [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

- [DSLRF] Rahrer, T., Ed., Fiandra, Ed., and Wright, Ed., "Triple-  
play Services Quality of Experience (QoE) Requirements",  
DSL Forum Technical Report TR-126, December 2006.
- [G.1082] ITU-T, "Measurement-based methods for improving the  
robustness of IPTV performance", ITU-T  
Recommendation G.1082, April 2009.
- [I-D.ietf-fecframe-interleaved-fec-scheme]  
Begen, A., "RTP Payload Format for 1-D Interleaved Parity  
FEC", draft-ietf-fecframe-interleaved-fec-scheme-09 (work  
in progress), January 2010.
- [I-D.ietf-fecframe-raptor]  
Waston, M., "Raptor FEC Schemes for FECFRAME",  
draft-ietf-fecframe-raptor-02 (work in progress),  
March 2010.
- [IEEE] IEEE, "Human Perception of Jitter and Media



Synchronization", IEEE Journal on Selected Areas in Communications Vol. 14, No. 1, January 1996.

[P.NAMS] ITU-T, "Non-intrusive parametric model for the Assessment of performance of Multimedia Streaming", ITU-T Recommendation P.NAMS, November 2009.

[RFC3357] Koodli, R. and R. Ravikanth, "One-way Loss Pattern Sample Metrics", RFC 3357, August 2002.

#### Authors' Addresses

Qin Wu  
Huawei  
101 Software Avenue, Yuhua District  
Nanjing, Jiangsu 210012  
China

Email: sunseawq@huawei.com

Glen Zorn  
Network Zen  
77/440 Soi Phoomjit, Rama IV Road  
Phra Khanong, Khlong Toie  
Bangkok 10110  
Thailand

Phone: +66 (0) 87 502 4274  
Email: gwz@net-zen.net

Roland Schott  
Deutsche Telekom Laboratories  
Deutsche-Telekom-Allee 7  
Darmstadt 64295  
Germany

Email: Roland.Schott@telekom.de



AVT Working Group  
Internet-Draft  
Intended status: Informational  
Expires: April 19, 2011

J. Xia  
Huawei  
October 16, 2010

Content Splicing for RTP Sessions  
draft-xia-avt-splicing-for-rtp-00

Abstract

This memo outlines RTP splicing. Splicing is a process that allows a new multimedia stream to be inserted into current multimedia stream and to be conveyed to receiver for a period of time. This memo discusses the requirements of RTP splicing. In order to satisfy the requirements, this memo lists several existing intermediary network elements as alternatives and evaluates whether one of alternatives can be used to perform RTP splicing.

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 April 19, 2011.

Copyright Notice

Copyright (c) 2010 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 . . . . .	3
2. Terminology . . . . .	3
3. RTP Splicing Topologies . . . . .	4
4. List of Alternatives for RTP Splicing . . . . .	7
4.1. Translator . . . . .	7
4.2. Mixer . . . . .	7
4.3. MCU . . . . .	8
5. Recommended Solution for RTP Splicing . . . . .	8
6. RTCP Sender Report Extensions . . . . .	10
7. Implementation considerations . . . . .	10
8. Security Considerations . . . . .	10
9. IANA Considerations . . . . .	11
10. Acknowledgments . . . . .	11
11. Normative References . . . . .	11
Author's Address . . . . .	12

## 1. Introduction

Splicing is a process that allows a new multimedia stream to be inserted into current multimedia stream and to be conveyed to receiver for a period of time. Splicing can be used for audio or video RTP streams.

One representative use case of using splicing is targeted advertisements (ads) insertion, which allows operators to override current program flow inserting its own targeted ads. So far [SCTE30] and [SCTE35] have standardized splicing for MPEG2-TS application, but to date there is no specification how to perform content splicing for RTP sessions [RFC3550].

In this document, we describe the topology of RTP splicing, and bring up the requirements of RTP splicing. Then we list several existing intermediary network elements as alternatives to handle RTP splicing and evaluate whether one of them can be off-the-shelf to satisfy the requirements on the aspect of feasibility, implementation complexity and backward compatibility.

## 2. Terminology

The keywords "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].

### Primary RTP Stream

A RTP stream RTP receiver is currently enjoying. A primary RTP stream is replaced by insertion RTP stream in part.

### Insertion RTP Stream

A RTP stream overrides primary RTP stream in part. Insertion RTP stream is output to RTP receiver for a period of time.

### Splicer

An intermediary node that inserts insertion RTP stream into primary RTP stream. Splicer sends insertion RTP stream to RTP receiver at the start of splicing, then switch back to primary RTP stream at the end of splicing.

### 3. RTP Splicing Topologies

In this document, we assume an intermediary network element, which is referred to as *Splicer*, receives primary RTP stream and insertion RTP stream(s), and outputs only one of these streams to RTP receiver at a point in time. The switch between the primary RTP stream and the insertion RTP stream(s) may be repeated during the RTP session. The RTP receiver receives only one stream at any point of time.

The splicing topology is depicted in Figure 1. The Splicer receives the primary RTP stream from media sender A and the insertion RTP stream from media sender C. Then Splicer selects one single RTP stream, either from A or from C, and outputs it to RTP receiver B and D over unicast or multicast paths. The criteria for stream selection are based on the policy from media sender A. How the policy is calculated is out of scope and not be discussed herein. The policy can be contained in extended RTCP SR sent from media source. Once Splicer receives RTCP SR packet, Splicer will learn how to splicing from the policy information, then strip the policy part from SR packet prior to forwarding the SR packet to RTP receiver. The specific RTCP SR extension are described in section 6.

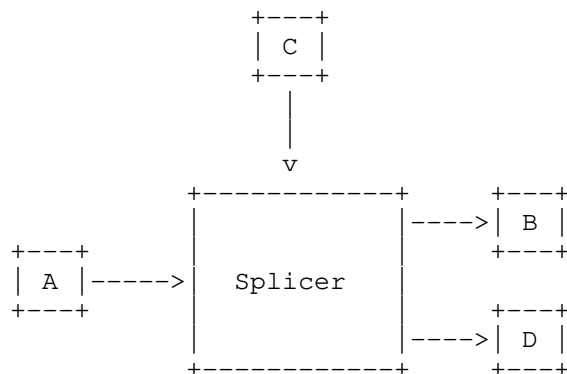


Figure 1: Splicing Topology

There may be more than one insertion RTP streams from different insertion sources. However, in order to facilitate the splicing process, we depict the simplest splicing stream flows scenario with only one insertion RTP stream in Figure 2.

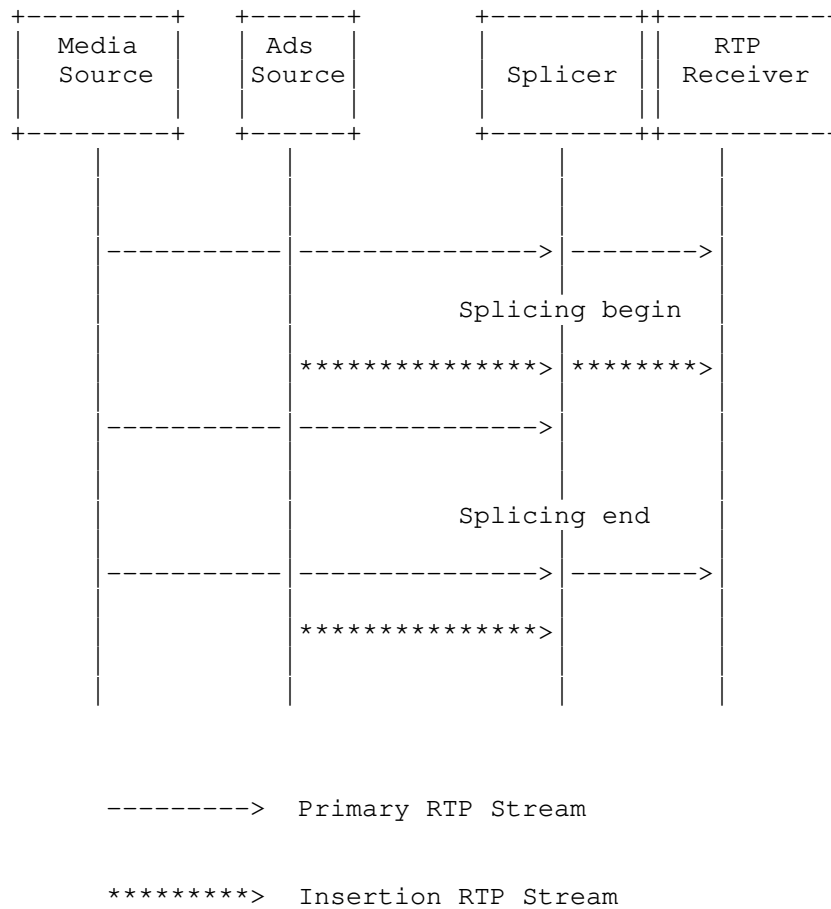


Figure 2: Splicing Stream Flows

1. When splicing begins, Splicer ceases forwarding primary RTP stream, and instead sends the insertion RTP stream to RTP receiver.
2. At the end of splicing, Splicer resumes the primary RTP stream and outputs the primary RTP packets to RTP receiver until next insertion RTP stream comes.

In order to guarantee a seamless splicing on RTP receiver, Splicer must carefully handle the joint of primary RTP stream and insertion RTP stream. In particular, Splicer needs to consider following requirements of RTP Splicing:

## REQ-1:

Splicer must be designed to operate in either unicast or multicast environment.

## REQ-2:

Splicer may receives multiple input streams, but must not synchronize or mix them into a single output stream. Instead, Splicer should select one of the multiple input streams and output it at any specific time.

## REQ-3:

To prevent the RTP receiver from easily identifying the inserted stream, the Splicer should merge the inserted stream into the primary RTP stream so that it will not be identified based on RTP header information like payload type number, sequence number and SSRC. This requires the Splicer to modify the RTP header of the inserted RTP stream. The Splicer may need to do media transcoding in most cases.

## REQ-4:

Splicer must be designed to guarantee RTP sequence number continuity in the case of switching from the primary RTP stream to the insertion RTP stream, and vice versa. Otherwise, a gap between the primary RTP stream and the insertion RTP stream may cause the RTP receiver to request retransmission for nonexistent packet loss while an overlap of the primary RTP stream and the insertion RTP stream may cause the RTP receiver to discard useful RTP packets due to the duplicate sequence numbers.

## REQ-5:

Splicer must be backward compatible with basic characteristics of [RFC3550], e.g., SSRC identifier collision resolution and loop detection.

## REQ-6:

Since the Splicer does RTP media transcoding on the inserted stream in most cases, Splicer should not simply forward RTCP traffic unaltered during the splicing.



#### 4. List of Alternatives for RTP Splicing

The basic RTP specification [RFC3550] explicitly supports the concept of Translator and Mixer, which are intermediary network elements that are involved in media transport functions. In addition, ITU-T audio-visual conferencing specification [H.323] defines the MCU (Multipoint Control Unit), which is also an intermediary network element that is used in video conference scenario.

In order to clarify the terms of Translator, Mixer and MCU, RTP topologies specification [RFC5117] specifically enumerates the different topologies and discusses the properties of each topology.

In this section, three alternatives (i.e., Translator, Mixer and MCU) based on [RFC5117] topologies would be evaluated. The following subsections will analyze which alternative can appropriately satisfy the requirements of Splicer.

##### 4.1. Translator

Transport Translator (Topo-Trn-Translator) does not modify the media stream itself, but are concerned with transport parameters. Transport Translator forwards RTP packets with their RTP header information intact and simply forwards RTCP packets unmodified as well.

Transport Translator does not satisfy the REQ-3, 4 and 6.

Media Translator (Topo-Media-Translator), in contrast, modifies the media stream itself. The modification of the media stream can be a full transcoding utilizing a different media codec, thus change the media format and timestamp. In most cases, Media Translator needs only to assign new sequence numbers to the outgoing RTP packets to guarantee the RTP sequence number continuity. Just like Transport Translator, Media Translator always keeps the SSRC identifier intact for any RTP stream across the translator.

Meanwhile, Media Translator cannot directly forward RTCP packets corresponding to the transcoded stream, and will need to insert itself into the RTCP flow, acting as a proxy for the RTP receiver.

Media Translator seems to dissatisfy REQ-3 (i.e., SSRC inconsistent).

##### 4.2. Mixer

Mixer (Topo-Mixer) aggregates multiple RTP streams, mixing them together to generate a new RTP stream. From media sender viewpoint, Mixer plays receiver role and terminates RTP streams sent from media

sender. While from RTP receiver viewpoint, Mixer plays media sender role and transmits the mixed RTP stream to RTP receiver with Mixer's own SSRC identifier. To facilitate loop detection, Mixer inserts a list of SSRC identifiers of the multiple input RTP streams into the CSRC identifiers field of the new output RTP stream.

Mixer is responsible for generating RTCP packets in accordance with its role. It is a RTP receiver and should therefore send Receiver Reports for the media streams it receives. In its role as a media sender, it should also generate Sender Reports for those media streams sent. More details are described in section 7.3 of [RFC3550].

Mixer does not satisfy the REQ-2.

#### 4.3. MCU

Video Switching MCU (Topo-Video-switch-MCU) forwards to RTP receiver a single RTP stream, selected from the multiple input RTP streams at a time. The forwarded stream changes during the session. Video Switching MCU may also perform media translation to modify the content in bit-rate, encoding, and timestamp. However, it still may indicate the original sender of the content through the SSRC.

Video Switching MCU only forwards RTCP Sender Reports for the currently selected media sender. Other RTCP processing behaviors are similar with Media Translator's.

Video Switching MCU does not satisfy the REQ-4 and REQ-3 (i.e., SSRC inconsistent).

RTCP-Terminating MCU (Topo-RTCP-terminating-MCU) limits each sender and receiver runs an RTP point-to-point session between itself and RTCP-Terminating MCU. The main feature of RTCP-Terminating MCU is that the SSRC identifiers of the media senders, whose content is included in the Mixer's output, is not indicated in CSRC identifiers list fields of the output RTP stream.

RTCP-Terminating MCU terminates RTCP traffic due to point-to-point topology.

RTCP-Terminating MCU does not satisfy the REQ-1 and 5.

#### 5. Recommended Solution for RTP Splicing

From the analysis of section 4, it seems that none of the alternatives perfectly satisfies the requirements of Splicer.

However, the topology of Media Translator describes a common translator for primary RTP stream and insertion RTP stream in section 4.1. In fact, an off-the-shelf Media Translator, which only translates primary RTP stream while terminating insertion RTP stream, can well satisfy all the requirements of Splicer. Media Translator terminates the insertion RTP stream whose SSRC does not affect the translator since the insertion RTP stream is a separate stream to the primary RTP stream.

When splicing begins, Media Translator transcodes inserted media codec into primary media codec and uses primary media source SSRC identifier in transcoded RTP stream to prevent the RTP receiver from easily identifying the inserted stream. Media Translator also assigns new sequence numbers to the inserted RTP packets. Note that the new sequence numbers of inserted RTP packets must seamlessly follow the sequence of the previous primary RTP packets before splicing.

When splicing ends, Media Translator switches back to primary RTP stream. During the splicing, the number of inserted RTP packets is unlikely to equal the number of overridden primary RTP packets because of media transcoding and different entropy coding. This requires Media Translator to modify the sequence numbers of subsequent primary RTP packets rather than directly forwarding them to RTP receiver. Note that the new sequence numbers of subsequent primary RTP packets must seamlessly follow the sequence of the last inserted RTP packet.

In this mode, RTP receiver does not need any RTP/RTCP extension for splicing, so there are not any serious backward compatibility issues. In contrast, it places the burden of performing splicing on Splicer. In the event that no insertion RTP stream is coming, the Splicer still decodes and re-encodes the primary RTP stream, recalculates the UDP/IP checksum and originates RR or SR messages in accordance with its role. In such case, overhead could be induced on Splicer compared to just forwarding the primary RTP stream to RTP receiver. If Splicer serves multiple primary RTP streams simultaneously, this may lead to worse overhead on Splicer.

Because insertion RTP stream is terminated on Media Translator, this requires Media Translator to generate its own RTCP Receiver Reports for the insertion RTP stream.

For the primary RTCP messages, Media Translator needs to interpose itself into RTCP SR, obtaining splicing information from RTCP SR extension and stripping the extension prior to forwarding the SR to RTP receiver.

[TBD: How to modify RTCP RR on translator is unsolved]

When receiving retransmission request for packet loss recovery during splicing, Media Translator first determines the range of lost packets requested by RTP receiver. If the sequence numbers of lost packets are in the range of inserted stream, Media Translator must initiate retransmission request messages on behalf of the RTP receiver toward corresponding retransmission server; Otherwise, Media Translator needs to modify the sequence numbers of the re-encoded primary lost packets prior to forwarding retransmission request to corresponding retransmission server.

## 6. RTCP Sender Report Extensions

[TBD]

## 7. Implementation considerations

When using Media Translator to handle RTP splicing, RTP receiver does not need any RTP/RTCP extension for splicing, so there are not any serious backward compatibility issues. In contrast, it places the burden of performing splicing on Media Translator. In the event that no insertion RTP stream is coming, the Media Translator still needs to decode and re-encode the primary RTP packets, to recalculate the UDP/IP checksum and originate RTCP messages. As a trade-off, overhead could be induced on Media Translator compared to just forwarding the primary RTP stream to RTP receiver. If Media Translator serves multiple primary RTP streams simultaneously, this may lead to worse overhead on Media Translator.

## 8. Security Considerations

The introduction of Media Translator must not diminish the level of security provided in current RTP/RTCP model. Since Media Translator alters the RTP payload, thus preventing the use of end-to-end encryption and source authentication, the Media Translator must be designed as a trusted device to take part in security context, e.g., support SRTP/SRTCP defined in [RFC3711] specification.

If encryption is employed, the Media Translator needs to decrypt the inbound media, as well as re-encrypt the outbound media. This requires Media Translator to set up different security associations with media senders and RTP receivers respectively.

## 9. IANA Considerations

[TBD]

## 10. Acknowledgments

[TBD]

## 11. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [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.
- [RFC4585] 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.
- [RFC5117] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 5117, January 2008.
- [RFC5760] Ott, J., Chesterfield, J., and E. Schooler, "RTP Control Protocol (RTCP) Extensions for Single-Source Multicast Sessions with Unicast Feedback", RFC 5760, February 2010.
- [SCTE30] Society of Cable Telecommunications Engineers (SCTE), "Digital Program Insertion Splicing API", 2001.
- [SCTE35] Society of Cable Telecommunications Engineers (SCTE), "Digital Program Insertion Cueing Message for Cable", 2004.
- [H.323] ITU-T Recommendation H.323, "Packet-based multimedia communications systems", June 2006.

Author's Address

Jinwei Xia  
Huawei  
Hui Hong Mansion  
Nanjing, Baixia District 210001  
China

Phone: +86-025-86622310  
Email: xiajinwei@huawei.com



Audio/Video Transport  
Internet-Draft  
Intended status: Standards Track  
Expires: April 21, 2011

P. Yang  
R. Even  
Huawei Technologies Co., Ltd.  
H. Moustafa  
France Telecom - Orange  
October 18, 2010

Switching from unicast to multicast for multicast short time-shift  
draft-yang-avt-switch-multicast-short-timeshift-00.txt

## Abstract

When a client requests a time-shift service for a multicast session using RTP for media transport, like pause, instant replay, slow-motion video, frame-by-frame viewing, rewind, fast-forward, etc., it needs to switch from multicast session to unicast session. This unicast session will always serve for the time-shift service unless the client manually switches back to the multicast session. Actually a time-shift request may happens for all clients. That is, multicast session stream will be replaced by many unicast time-shift streams having significant impact on network bandwidth usage.

In this document, we describe a method using existing RTP and RTCP protocol infrastructure that switches back to multicast session from unicast session of time-shift. In this method, a burst RTP flow which is a little faster than the primary multicast rate may be transmitted so that unicast session may catch up and switch back to multicast session.

## 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 April 21, 2011.

## Copyright Notice



Copyright (c) 2010 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 . . . . .	3
1.1. RTSP Time-shift and MST Reverse Switch . . . . .	5
1.2. RAMS and MST Reverse Switch . . . . .	6
2. Conventions . . . . .	6
3. Definitions . . . . .	6
4. Reverse switch for multicast short time-shift . . . . .	7
4.1. Overview . . . . .	7
4.2. Message Flows . . . . .	8
5. Security Considerations . . . . .	10
6. IANA Considerations . . . . .	10
7. Acknowledgements . . . . .	10
8. Normative References . . . . .	10
Authors' Addresses . . . . .	10

## 1. Introduction

Interactive time-shift is an important service for media application. Through time-shift control operations (e.g. pause, instant replay, slow-motion video, frame-by-frame viewing, rewind and fast-forward), viewers can access recorded programming and live streams. The time-shift service is available for recorded media and real-time streaming. For the real-time streaming, time-shift service requires recording/caching of the information. The time-shifted information can be sent by the original media source or by a server in the network that caches the stream and provides the time-shift service.

Note that time-shift service may run on the client side but this requires the client to be able to cache the stream and synchronization to the main multicast stream needs to be managed by the client and does not require any specific protocol.

This document looks at time-shift services where the original content is delivered using multicast. In this use case the time-shift service is using a unicast stream from a Multicast Short Time-shift (MST) server to the client. The synchronization between the multicast and unicast stream can be achieved by the client if he is using control commands like fast forward or RTSP [ref] Scale request. The MST can notice that the client is receiving in the unicast stream the same information that is current in the multicast stream.

Network traffic will rise with the increase in the number of clients using time-shift service, because the time-shift service traffic changes from a multicast stream (one multicast stream for all clients) to a large number of unicast streams (one unicast stream per client). Time-shift services like slow-motion view, instant replay, frame-by-frame viewing, pause and rewind may often occur for most of viewers in prime-time (e.g. when watching sports events).

We can separate the time-shift services of primary live streams delivered over multicast/broadcast into multicast long time-shift and multicast short time-shift. Some of the manipulations requested by the viewers are multicast short time-shift (e.g. short pause, instant replay, slow-motion video and frame-by-frame viewing). For the multicast short time-shift, the time offset between the unicast time-shift playback, after the time-shift request, and the current time of the primary multicast stream is small enabling the unicast stream to catch up with primary multicast stream by speeding up the playback, and then switch back to the original multicast session.

In this document, we propose a solution for enabling the time-shifted stream receivers to catch up with the original multicast stream using the tools offered by the existing RTP and RTCP protocols. The

document also describes how to switch back to the primary multicast session from the unicast session initiated by either the client or the server. Using this solution allows the network bandwidth to be effectively saved for the time-shift service of multicast application.

In the scenario that we consider in this draft, an intermediary network element (that we refer to as Multicast Short Time-shift server) joins the multicast session and continuously caches the multicast stream. When an RTP receiver sends a time-shift request, the MST server starts a new unicast RTP session and transmits the unicast stream to the RTP receiver over that session. A simplified network diagram showing this scenario employing an intermediary network element is shown in Figure 1, where the hashed lines show the unicast stream.

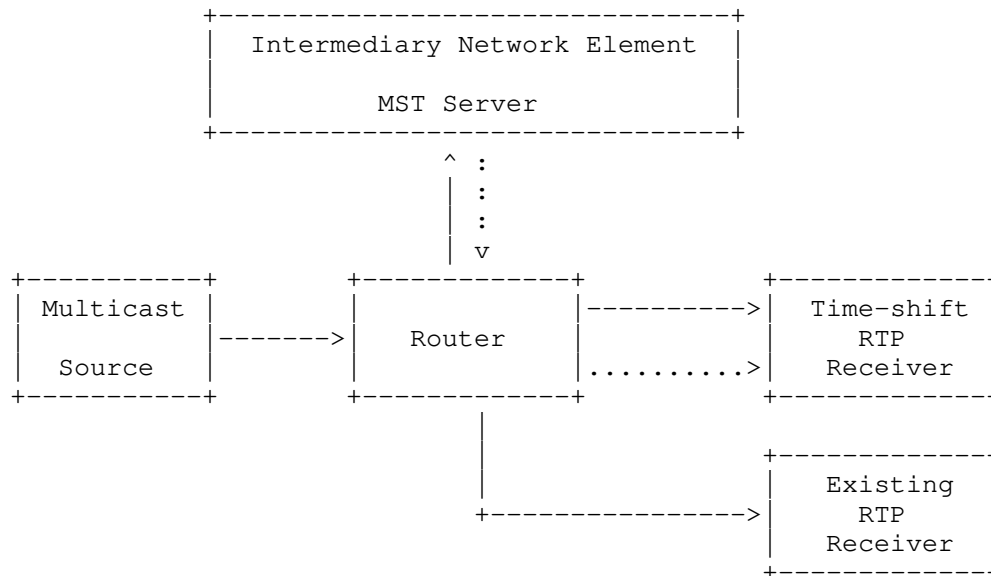


Figure 1: Reverse switch for multicast short time-shift through an intermediary network element

The proposed solution is not dependent on a specific streaming control protocol like RTSP [ref]. It addresses the synchronization between a primary multicast RTP stream and a parallel time-shifted unicast RTP stream. A principle design goal is to use an existing protocol to define this solution. This improves the versatility of the existing implementations, and promotes faster deployment and better interoperability. To this effect, the proposal is to use the switching of flows from unicast stream to multicast stream described in RAMS [draft-ietf-avt-rapid-acquisition-for-rtp], and use the capabilities of RTCP to handle the signaling needed to accomplish

this automatic reverse switch for multicast short time-shift.

#### 1.1. RTSP Time-shift and MST Reverse Switch

RTSP 2.0[draft-ietf-mmusic-rfc2326bis] is an application-level protocol for setup and control of the media delivery with real-time properties, and provides an extensible framework to enable controlled, on-demand delivery of real-time data, typically streaming media.

RTSP defines any necessary media transport signalling with regards to RTP. In appendix C.1 it defines the interaction of RTSP with respect to the RTP protocol [RFC3550]. Yet RTSP is not limited to RTP media delivery control but any delivery type of data. It provides a general media delivery control mechanism to play out the media, pause media, or stop media over different delivery channels, such as UDP, multicast UDP, TCP, RTP over UDP, etc.

RTSP can also provide interactive time-shift function with different scale and speed. Section 13.4.8 in [draft-ietf-mmusic-rfc2326bis] has a scenario for Playing Live with time-shift where a certain media server may offer time-shift services to their clients. The usage of this play method can implement time-shift for Live Media or On-demand Media. Section 16.44 in [draft-ietf-mmusic-rfc2326bis] addresses the scaling for video and audio, it suggests sending only key frames for video but this may not achieve the right scale speed since it depends on the locality of such frames in the stream. It also requires analysis of the media payload for all supported codecs to find the key frames. In this document we call the interactive time-shift function in [draft-ietf-mmusic-rfc2326bis] as "RTSP Time-shift".

For Live Media using RTP multicast over UDP, RTSP Time-shift can provide an initial switch from multicast session to unicast session when time-shift happens, but it cannot complete the reverse switch from unicast session of time-shift to the original multicast session.

An MST server can for example transmit the unicast stream of the time-shift by a short burst stream and indicate to the receivers to speed up the playback through Non-perceptual Speedup Playback. After the burst of the unicast stream of time-shift catches up with the primary multicast stream, the client can switch back to the primary multicast stream so that the network bandwidth can be effectively saved. The synchronization due to catch up with the primary multicast session may also happen due to the client operation like the fast forward.

## 1.2. RAMS and MST Reverse Switch

RAMS [draft-ietf-avt-rapid-acquisition-for-rtsp] describes a method using the existing RTP and RTCP protocol for reducing the acquisition delay, allowing fast channel switch. RAMS defines the reverse switch from unicast burst session to the primary multicast session.

## 2. Conventions

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

This document uses the following acronyms and definitions frequently:

**Time-shift Control:** Refers to the manipulation control of time-shift service (e.g. pause, slow-motion video, frame-by-frame viewing, rewind, fast-forward, etc.), not include the play method.

**Time-shift Playback:** Refers to the normal playback with a natural playback rate after the time-shift control.

**Multicast Time-shift Interval:** Refers to the time interval or the number of packets between the primary multicast stream and the normal playback during the unicast session after the time-shift control. In other words, it presents the time interval between the start point frame of the unicast session and the frame of the current location of the primary multicast session.

**Multicast Short Time-shift:** The case when the multicast Time-shift Interval is short, such as 30 seconds. The normal playback during the unicast session after the time-shift control means that viewers finish the time-shift control (e.g. pause, rewind and fast-forward) and start to receive the unicast stream and play back again at the speed of the Primary multicast stream.

**Primary multicast stream:** The multicast stream before time-shift request.

**Instant Replay:** It is the case of the replaying of video footage of an event or incident directly after its occurrence. In television broadcasting of sports events, instant replay is often used during live broadcast, to show a passage of play which was important or remarkable, or which was unclear on the first sight.

**Slow-motion Video:** The case when the playback of a video clip appears to be slower than the natural speed of the events.

**Non-perceptual Speedup Playback:** During the speedup playback, after each interval of some frames, one frame is skipped as if it was not present, the next frame is directly rendered to take the place of the skipped frame, while keeping a fixed Frame Per Second (FPS) during the playback speedup.

#### 4. Reverse switch for multicast short time-shift

This section gives an overview on the proposed method for reverse multicast time- shift switch from unicast time-shift RTP Sessions.

##### 4.1. Overview

RTP Control Protocol (RTCP) Extensions for Single-Source Multicast Sessions with Unicast Feedback [RFC5760] specifies an extension to the RTCP to use unicast feedback to a multicast sender. The (Unicast RTCP) Feedback Target is a logical function to which RTCP unicast feedback traffic is addressed. The functions of the Feedback Target and the Distribution Source MAY be co-located or integrated in the same entity. In this case, The (Unicast RTCP) Feedback Target MAY be co-located or integrated in the Multicast Time-shift Server.

This section presents a proposed method to switch back to the multicast session from the unicast session of time-shift considering multicast short time-shift. The proposed method allows the network bandwidth to be effectively saved when an RTP receiver finishes its time-shift request and starts its time- shift playback by unicast stream. A Multicast Short Time-shift Server (MST) and new RTCP feedback messages are also introduced for the proposed method.

The MST server has a cache where the most recent packets from the primary multicast stream are continuously stored for some time. When a viewer wants to normally playback the unicast stream after time-shift request, the RTP receiver needs to send a playback request to the feedback target, and then receives the unicast stream from the MST server. In order to switch back to the original multicast stream, the MST server needs to transmit a burst unicast stream to RTP receivers. Using an accelerated bitrate (as compared to the bitrate of the original multicast stream). This means that at a certain point in time, the unicast burst will catch up with the original multicast stream. At this point, the RTP receiver no longer needs to receive the unicast burst and can switch back to the original multicast session.

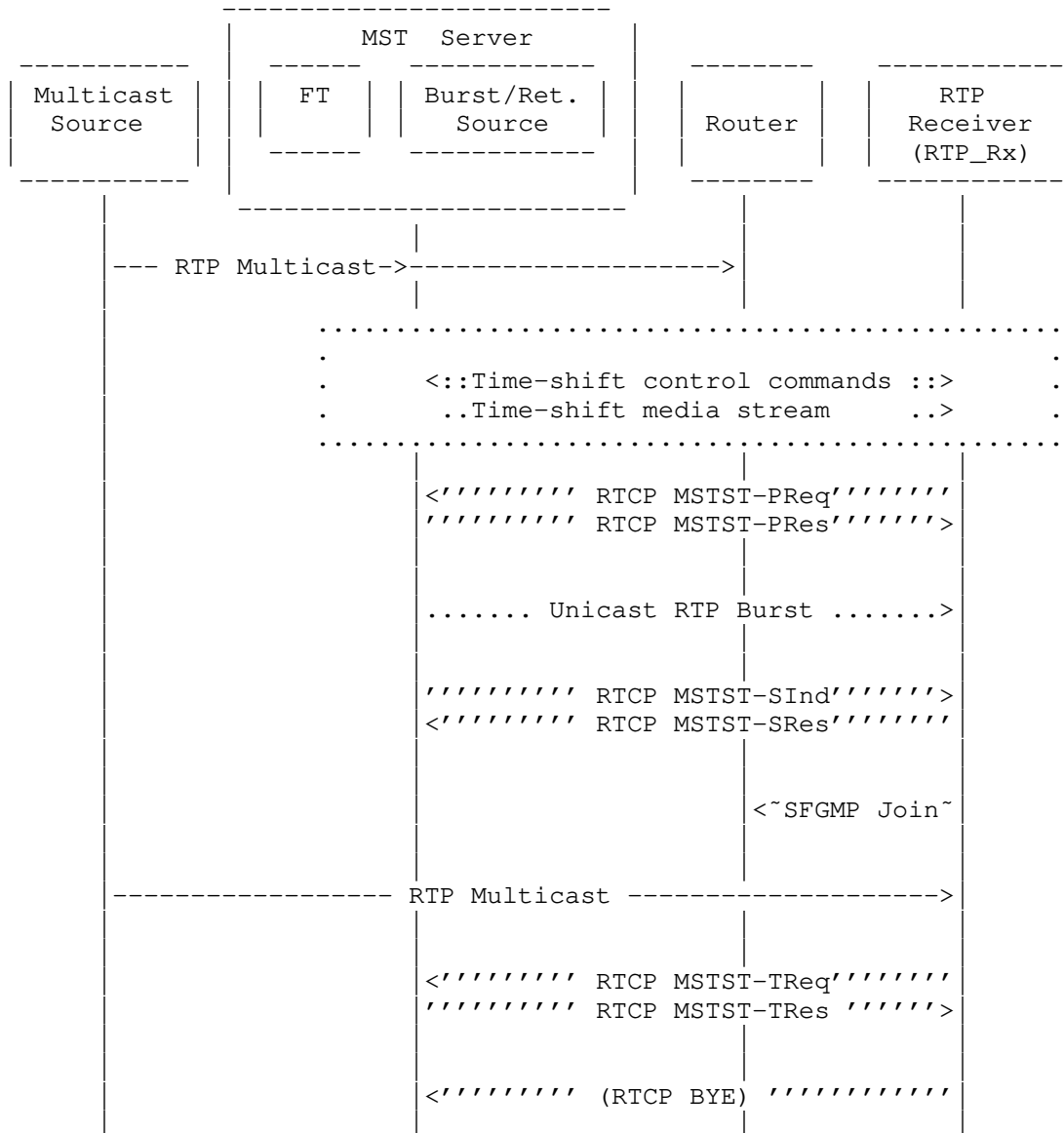
The transmission of the unicast burst stream of time-shift playback depends on the time-shift buffer size of RTP receivers and the Multicast Short Time-shift Interval. In the case when the time-shift buffer size of an RTP receiver can inadequately accommodate the number of packets of the Multicast Short Time-shift Interval, the receiver needs to playback for the duration of the speeding-up until the remaining number of packets does not overflow at the time-shift buffer. During the playback of the speeding-up, the receiver may speed up video playback by the non-perceptual speedup playback so that viewers could never perceived the existence of the speeding-up. Refer to [I-D.yang-avt-rtp-synced-playback]

#### 4.2. Message Flows

This section introduces the different messages for the proposed method and the related messages flows.

- MST(Multicast Short Time-shift) synchronization transmission(MSTST): This is the generic form of the different messages that will be transmitted during the proposed method in this draft for reverse switch multicast short time-shift.
- Time-Shift-Control-Commands: This message presents the request for the Time-Shift service (e.g. fast forward, pause, reverse, ..etc.). This message may be addressed by the MST server or other time-shift server.
- MSTST Playback Request(MSTST-PReq): It's the message sent from the RTP receiver to the MST to request the playback after the time-shift request.
- MSTST Playback Response(MSTST-PRes): It's the response message sent from the MST to the RTP receiver to confirm the playback process initiation. Following this message, the RTP receiver will receive the unicast burst RTP stream from the MST.
- MSTST Synchronization Indication(MSTST-SInd): When the unicast burst catch up with the original multicast stream, this message is sent from the MST to the RTP receiver to indicate this fact.
- MSTST Synchronization Response(MSTST-SRes): This message is sent by the RTP receiver as a response to the MST confirming the need to switch back to the original multicast session. Following this message the RTP receiver is able to receive the multicast stream.
- MSTST Termination Request(MSTST-TReq): This message is sent by the RTP receiver to the MST to terminate the process of the time-shift.

- **MSTST Termination Response(MSTST-TRes)**: This message is a reply sent by the MST to the RTP to confirm to the MSTST-TReq.





5. Security Considerations

TBD.

6. IANA Considerations

TBD.

7. Acknowledgements

TBD.

8. Normative References

[I-D.ietf-mmusic-rfc2326bis]

Schulzrinne, H., Rao, A., Lanphier, R., Westerlund, M.,  
and M. Stiemerling, "Real Time Streaming Protocol 2.0  
(RTSP)", draft-ietf-mmusic-rfc2326bis-24 (work in  
progress), July 2010.

[I-D.yang-avt-rtp-synced-playback]

Yang, P. and Y. Wang, "Synchronized Playback in Rapid  
Acquisition of Multicast Sessions",  
draft-yang-avt-rtp-synced-playback-04 (work in progress),  
March 2010.

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

[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

Peilin Yang  
Huawei Technologies Co., Ltd.  
101 Software Avenue, Yuhua District, Nanjing 210012  
P.R.China

Phone: +86-25-56622638  
Email: yangpeilin@huawei.com

Roni Even  
Huawei Technologies Co., Ltd.  
14 David Hamelech, Tel Aviv 64953  
Israel

Email: even.roni@huawei.com

Hassnaa Moustafa  
France Telecom - Orange  
38-40 rue du General Leclerc Issy Les Moulineaux, 92794 Cedex 9  
France

Email: hassnaa.moustafa@orange-ftgroup.com

Tina Tsou (editor)  
Huawei Technologies  
Section F, Huawei Industrial Base  
Bantian Longgang, Shenzhen 518129  
P.R. China

Phone: +86 755 28972912  
Email: tena@huawei.com

Gu Yingjie  
Huawei Technologies Co., Ltd.  
101 Software Avenue, Yuhua District, Nanjing 210012  
P.R.China

Email: guyingjie@huawei.com

