VIDEO SERVICES FORUM

Preamble to Video Services Forum (VSF) Technical Recommendation TR-06-1:2020

June 25, 2020

The Reliable Internet Stream Transport (RIST) project was initiated as an Activity Group under the auspices of the Video Services Forum in 2017. To date, the group has produced two specifications, released as TR-06-1 (RIST Simple Profile) and TR-06-2 (RIST Main Profile). Work continues within the group towards developing additional RIST specifications that include additional features. As the Activity Group develops and reaches consensus on new functions and capabilities, these documents will also be released in support of the RIST effort.

The attached document is a revision of the original TR-06-1 RIST Simple Profile Specification. The only difference between the original 2018 version of Simple Profile and the current document is the addition of the optional RTT Echo message, documented in section 5.2.6, and some changes in language to be more accurate about the normative and informative provisions of the document. The purpose of this optional message is to provide a mechanism whereby a RIST receiver can measure the round-trip time between itself and the RIST sender. This information may be used by the RIST receiver to optimize its NACK requests, as network conditions change.

The RTT Echo Request/Response messages are implemented as RIST APP RTCP packets with new subtypes. Such messages will be silently discarded by RIST devices that do not implement this functionality, thus ensuring interoperability.

RIST Simple Profile devices are manually configured with a static value for the NACK window. In the absence of any additional information, this static value is used in managing retransmission requests. Therefore, a RIST receiver that sends RTT Echo messages and does not receive a response will simply continue to operate using this static value.

The primary purpose of RIST Simple Profile is to ensure reliable transmission over the Internet. The RTT Echo message represents an optional mechanism in support of that function. Therefore, the Activity Group elected add this message to Simple Profile, and revise TR-06-1.

For additional information about the RIST Activity group, or to find out about participating in the development of future specifications, please visit http://vsf.tv/RIST.shtml

Video Services Forum (VSF)
Technical Recommendation TR-06-1

Reliable Internet Stream Transport (RIST) Protocol Specification – Simple Profile

June 25, 2020

VSF_TR-06_2020_06_25

INTELLECTUAL PROPERTY RIGHTS

THE FORUM DRAWS ATTENTION TO THE FACT THAT IT IS CLAIMED THAT COMPLIANCE WITH THIS RECOMMENDATION MAY INVOLVE THE USE OF A PATENT ("IPR") CONCERNING SECTIONS 4, 5 , 5.1 (EXCLUDING 5.1.2), 5.2 , 5.3 AND ITS SUB SECTIONS, AND 5.4.

THE FORUM TAKES NO POSITION CONCERNING THE EVIDENCE, VALIDITY OR SCOPE OF THIS IPR.

THE HOLDER OF THIS IPR HAS ASSURED THE FORUM THAT IT IS WILLING TO LICENSE ALL IPR IT OWNS AND ANY THIRD PARTY IPR IT HAS THE RIGHT TO SUBLICENSE WHICH MIGHT BE INFRINGED BY ANY IMPLEMENTATION OF THIS RECOMMENDATION TO THE FORUM AND THOSE LICENSEES (MEMBERS AND NON-MEMBERS ALIKE) DESIRING TO IMPLEMENT THIS RECOMMENDATION. INFORMATION MAY BE OBTAINED FROM:

VIDEO-FLOW.LTD
11 HA'AMAL ST. ROSH HA'AYIN ISRAEL, 4809241

ATTENTION IS ALSO DRAWN TO THE POSSIBILITY THAT SOME OF THE ELEMENTS OF THIS RECOMMENDATION MAY BE THE SUBJECT OF IPR OTHER THAN THOSE IDENTIFIED ABOVE. THE FORUM SHALL NOT BE RESPONSIBLE FOR IDENTIFYING ANY OR ALL SUCH IPR.

THIS RECOMMENDATION IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NONINFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY USE OF THIS RECOMMENDATION SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER THE FORUM, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY MPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER, DIRECTLY OR INDIRECTLY, ARISING FROM THE USE OF THIS RECOMMENDATION.

LIMITATION OF LIABILITY

VSF SHALL NOT BE LIABLE FOR ANY AND ALL DAMAGES, DIRECT OR INDIRECT, ARISING FROM OR RELATING TO ANY USE OF THE CONTENTS CONTAINED HEREIN, INCLUDING WITHOUT LIMITATION ANY AND ALL INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES (INCLUDING DAMAGES FOR LOSS OF BUSINESS, LOSS OF PROFITS, LITIGATION, OR THE LIKE), WHETHER BASED UPON BREACH OF CONTRACT, BREACH OF WARRANTY, TORT (INCLUDING NEGLIGENCE), PRODUCT LIABILITY OR OTHERWISE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THE FOREGOING NEGATION OF DAMAGES IS A FUNDAMENTAL ELEMENT OF THE USE OF THE CONTENTS HEREOF, AND THESE CONTENTS WOULD NOT BE PUBLISHED BY VSF WITHOUT SUCH LIMITATIONS.

Executive Summary

Many solutions exist in the market for reliable streaming over the Internet. These solutions all use the same types of techniques, but they are all proprietary and do not interoperate. This Technical Recommendation (TR) contains a protocol specification for reliable streaming over the Internet, so end users can mix and match solutions from different vendors.

Recipients of this document are requested to submit notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the Recommendation set forth in this document, and to provide supporting documentation.

This TR-06-1:2020 supersedes TR-06-1:2018. This new version adds the RTCP RTT Echo Request command described in Section 5.2.6 of the document. Note that this added function is optional, but recommend for new and existing implementations.

Table of Contents

1 Introduction (Informative)

As broadcasters and other video users increasingly utilize unconditioned Internet circuits to transport high-quality video, the demand grows for systems that can compensate for the packet losses and delay variation that often affect these streams. A variety of solutions are currently available on the market, however, incompatibilities exist between devices from different suppliers.

The Reliable Internet Stream Transport (RIST) project was launched specifically to address the lack of compatibility between devices, and to define a set of interoperability points through the use of existing or new standards and recommendations.

1.1 Contributors

The following individuals participated in the Video Services Forum RIST working group which developed this technical recommendation.

Merrick Ackermans (MVA Broadcast Consulting)	Sergio M Ammirata (SipRadius)	Paul Atwell (Media Transport Solutions)
Uri Avni (Zixi)	John Beer (QVidium)	Ghislain Collette (Haivision)
Magnus Danielson (NetInsight)	Israel Drori (Zixi)	Eric Fankhauser (Evertz)
Ronald Fellman (QVidium)	Michael Firth (Nevion)	Rafael Fonseca (Artel)
Oded Gants (Zixi)	Peter Keys (Charter Communications)	Holger Klaas (Nevion)
Brian Matherly (Sencore)	Ciro Noronha (Cobalt Digital)	Andy Rayner (Nevion)
Steve Riedl (Turner)	David Robison (CenturyLink)	Adi Rozenberg (VideoFlow)
Bob Ruhl (VSF)	Manjinder Sandhu (Evertz)	Wes Simpson (Telecom Product Consulting)
Mikael Wånggren (Net Insight)	Adam Yellen (Haivision)	

1.2 About the Video Services Forum

The Video Services Forum, Inc. (www.videoservicesforum.org) is an international association dedicated to video transport technologies, interoperability, quality metrics and education. The VSF is composed of service providers, users and manufacturers. The organization's activities include:

- providing forums to identify issues involving the development, engineering, installation, testing and maintenance of audio and video services;
- exchanging non-proprietary information to promote the development of video transport service technology and to foster resolution of issues common to the video services industry;
- identification of video services applications and educational services utilizing video transport services;
- promoting interoperability and encouraging technical standards for national and international standards bodies.

The VSF is an association incorporated under the Not For Profit Corporation Law of the State of New York. Membership is open to businesses, public sector organizations and individuals worldwide. For more information on the Video Services Forum or this document, please call +1 929-279-1995 or e-mail opsmgr@videoservicesforum.org.

2 Conformance Notation

Normative text is text that describes elements of the design that are indispensable or contains the conformance language keywords: "shall", "should", or "may". Informative text is text that is potentially helpful to the user, but not indispensable, and can be removed, changed, or added editorially without affecting interoperability. Informative text does not contain any conformance keywords.

All text in this document is, by default, normative, except the Introduction and any section explicitly labeled as "Informative" or individual paragraphs that start with "Note:"

The keywords "shall" and "shall not" indicate requirements strictly to be followed in order to conform to the document and from which no deviation is permitted.

The keywords, "should" and "should not" indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.

The keywords "may" and "need not" indicate courses of action permissible within the limits of the document.

The keyword “reserved” indicates a provision that is not defined at this time, shall not be used, and may be defined in the future. The keyword “forbidden” indicates “reserved” and in addition indicates that the provision will never be defined in the future.

A conformant implementation according to this document is one that includes all mandatory provisions ("shall") and, if implemented, all recommended provisions ("should") as described. A conformant implementation need not implement optional provisions ("may") and need not implement them as described.

Unless otherwise specified, the order of precedence of the types of normative information in this document shall be as follows: Normative prose shall be the authoritative definition; Tables shall be next; followed by formal languages; then figures; and then any other language forms.

3 References

SMPTE ST 2022-1:2007, Forward Error Correction for Real-Time Video/Audio Transport Over IP Networks

SMPTE ST 2022-2:2007, Unidirectional Transport of Constant Bit Rate MPEG-2 Transport Streams on IP Networks

SMPTE ST 2022-7:2013, Seamless Protection Switching of SMPTE ST 2022 IP Datagrams

IETF RFC 3550, RTP: A Transport Protocol for Real-Time Applications

IETF RFC 3551, RTP Profile for Audio and Video Conferences with Minimal Control

IETF RFC 4585, Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVFP)

4 RIST Profiles

RIST will have multiple operational profiles, corresponding to increasing levels of complexity and functionality. Higher profiles will include all the features and functionality of the preceding profiles. This document defines RIST Simple Profile. The attached document, *Preamble to Video Services Forum (VSF) Technical Recommendation TR-06-1:2020* provides an overview of the TR-06 RIST work to date and describes additional work in progress as of April, 2020.

5 Simple Profile

RIST Simple Profile provides only basic interoperability and packet loss recovery. All configuration is manual and done outside the protocol.

5.1 Baseline Protocol

In order to ensure a level of interoperability between RIST and non-RIST implementations, RTP shall be used as the baseline protocol for media transport.

If an RTP standard exists for a certain media type, that standard shall be used as the definition of the RTP header fields. For example, if the media to be transported is in the format of an MPEG-2 Transport Stream, SMPTE-2022-1/2 shall be used for the baseline stream.

Note: RIST will augment the baseline RTP transmission with mechanisms to recover from packet loss.

Feedback/control messages shall use RTCP, as specified in RFC 3550.

5.1.1 Unicast Port Assignments

In a unicast environment (such as most of the Internet), RIST senders transmit to the unicast IP addresses of the RIST receivers. In this environment, the following rules will apply:

1. RIST senders shall transmit the RTP media packets to the configured IP address of the RIST receiver and a user-selected UDP destination port P, where P is an even number between 2 and 65534.

   RIST receivers shall listen on UDP port P for media. This transmission is unidirectional, from sender to receiver.

   The sender may choose any arbitrary source port M for the RTP flow.

2. RIST senders shall periodically transmit the compound RTCP packets specified in section 5.2.1 to the configured IP address of the RIST receiver and UDP port P+1.

   The sender may choose any arbitrary source port R for the RTCP packets.

   RIST senders shall listen on port R for RTCP packets from the RIST receiver.

3. RIST receivers shall listen on UDP port P+1 for RTCP packets from the sender. The source IP address of such packets is denoted by S and their source UDP port is denoted by R'.

   Note: In the absence of NAT devices between the sender and receiver, R and R' will be the same.

   RIST receivers shall send the RTCP packets they generate to IP address S and destination UDP port R' of the RIST sender, with a source UDP port of P+1.

   A RIST receiver shall use S and R' from the last valid RTCP packet it has received from the RIST sender.

4. RIST senders may offer the user the ability to manually configure source ports M and R.

5. Receiving RIST devices may use UPnP to configure automatically firewalls.

Note: These choices simplify the interaction of the RIST senders and receivers with NAT firewalls, as follows:

- If the RIST sender is behind a NAT device, the outgoing RTP and RTCP packets establish state in the device, allowing the RTCP packets from the receiver to come back to the sender.
- If the RIST receiver is behind a NAT device, only ports P and P+1 need to be opened for operation.

5.1.2 Multicast Port Assignments

RIST can also be applied to multicast environments, such as private or isolated networks, or networks connected with multicast-capable tunnels. In a multicast environment, RIST shall follow the standard UDP port assignments as per RFC 3550:

1. RIST senders shall transmit the RTP media packets to a user-selected UDP destination port P, where P is an even number between 2 and 65534.

   RIST receivers shall listen on UDP port P for media. This transmission is unidirectional, from sender to receiver.

   The RIST sender may choose any arbitrary source port M for the RTP flow.

2. RIST senders shall periodically transmit the compound RTCP packets specified in section 5.2.1 to UDP port P+1, using the same multicast destination address as the media.

   The RIST sender may choose any arbitrary source port R for the RTCP packets.

   RIST senders shall listen on port P+1 for RTCP packets from the RIST receiver.

3. RIST receivers shall listen on UDP port P+1, using the same multicast IP address as the media for RTCP packets from the RIST sender.

4. RIST receivers shall send the RTCP packets they generate to the same multicast address as the media, and destination UDP port P+1.

   The RIST receiver may choose any arbitrary UDP source port for the RTCP packets it transmits.

5. RIST senders may offer the user the ability to configure manually the source ports M and R. RIST senders may use R=P+1 for simplicity.

5.2 RTCP Support

RIST senders and receivers shall implement a minimal subset of RTCP as described in this section. For senders, RTCP is used primarily to keep state on NAT devices along the path. For receivers, RTCP is used primarily to request lost packet retransmissions. The additional information included in the RTCP packet may be used sender and receiver devices to achieve better network performance.

Note: Section 5.2.6 describes optional RTCP RTT Echo request/response packets that can be used by receivers to compute the Round Trip Time (RTT) between themselves and the sender.

This information can be used by the receiver to optimize the timing of their retransmission requests, described in section 5.3.

5.2.1 Compound RTCP Packets

Multiple RTCP packets can be concatenated without any intervening separators and sent out in one UDP payload.

RIST senders and receivers shall always transmit compound RTCP packets in order to comply with RFC 3550. The combinations shall be:

- RIST Senders:
    - Sender Report (SR) Packet (section 5.2.2) **OR** Empty Receiver Report (RR) Packet (section 5.2.3)
    - Source Description (SDES) Packet with a CNAME field (section 5.2.5)
    - RTT Echo Request or Response Packets, if supported (section 5.2.6)
- RIST Receivers
    - Receiver Report (RR) Packet (section 5.2.4) or Empty Receiver Report (RR) Packet (section 5.2.3)
    - Source Description (SDES) Packet with a CNAME field (section 5.2.5)
    - NACK Packet (if required – section 5.3.2)
    - RTT Echo Request or Response Packets, if supported (section 5.2.6)

RIST senders and receivers shall transmit RTCP packets periodically. The rate at which the RTCP packets are sent shall comply with the following requirements:

1. The interval between two successive RTCP packets shall be 100 milliseconds or less.
2. The maximum data rate used by RTCP packets shall be no higher than 5% of the average media data rate. For very low bit rate applications, requirement one (1) above takes precedence.

The packet formats for all items are specified in the following sections.

5.2.2 Sender Report (SR) RTCP Packets

Since the primary use of RIST is in unicast links, the SR packet shall have no reception report blocks. The SR packet description presented below has been adapted from RFC 3550 section 6.4.1.

RIST senders shall use either the SR packet described in this section or an empty RR packet described later in this document.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|  RC=0   |   PT=SR=200   |           length=6            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         SSRC of sender                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              NTP timestamp, most significant word             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             NTP timestamp, least significant word             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         RTP timestamp                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     sender's packet count                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      sender's octet count                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The fields are used as follows:

version (V): 2 bits
: Identifies the version of RTP, which is the same in RTCP packets as in RTP data packets. RIST packets shall have V=2.

padding (P): 1 bit
: Indicates whether or not there is padding at the end of the packet. RIST SR packets shall have P=0.

reception report count (RC): 5 bits
: The number of reception report blocks contained in this packet. RIST SR packets shall have RC=0.

packet type (PT): 8 bits
: Contains the constant 200 to identify this as an RTCP SR packet.

length: 16 bits
: The length of this RTCP packet in 32-bit words minus one, including the header and any padding. RIST SR packets shall have length=6.

SSRC: 32 bits
: The synchronization source identifier for the originator of this SR packet.

NTP timestamp: 64 bits
: Indicates the wallclock time when this report was sent. The most significant 32 bits on this field indicate the number of seconds since 0h UTC on January 1900, and the least significant 32 bits indicate the fraction of the second. On a system

that has no notion of wallclock time but does have some system-specific clock such as "System uptime", a sender may use that clock as a reference to calculate relative NTP timestamps. A sender that has no notion of wallclock or elapsed time may set the NTP timestamp to zero.

RTP timestamp: 32 bits

Corresponds to the same time as the NTP timestamp (above), but in the same units and with the same random offset as the RTP timestamps in data packets. Note that in most cases this timestamp will not be equal to the RTP timestamp in any adjacent data packet. Rather, it shall be calculated from the corresponding NTP timestamp using the relationship between the RTP timestamp counter and real time as maintained by periodically checking the wallclock time at a sampling instant.

sender's packet count: 32 bits

The total number of RTP data packets transmitted by the sender since starting transmission up until the time this SR packet was generated. The count should be reset if the sender changes its SSRC identifier.

sender's octet count: 32 bits

The total number of payload octets (i.e., not including header or padding) transmitted in RTP data packets by the sender since starting transmission up until the time this SR packet was generated. The count should be reset if the sender changes its SSRC identifier.

5.2.3 Empty Receiver Report (RR) RTCP Packets

Since the primary purpose of the RTCP transmission from the sender to the receiver is to possibly establish state in the firewalls along the path, the sender may elect to send an empty RR instead of the SR defined in the previous section. The empty RR is depicted below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|   RC=0   |  PT=RR=201    |             length=1          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         SSRC of sender                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The fields are used as follows:

version (V): 2 bits

Identifies the version of RTP, which is the same in RTCP packets as in RTP data packets. RIST packets shall have V=2.

padding (P): 1 bit

Indicates whether or not there is padding at the end of the packet. RIST RR packets shall have P=0.

reception report count (RC): 5 bits

The number of reception report blocks contained in this packet. Empty RR packets shall have RC=0.

packet type (PT): 8 bits

Contains the constant 201 to identify this as an RTCP RR packet.

length: 16 bits

The length of this RTCP packet in 32-bit words minus one, including the header and any padding. Empty RR packets shall have length=1.

SSRC: 32 bits

The synchronization source identifier for the originator of this RR packet.

5.2.4 Receiver Report (RR) RTCP Packets

RIST RR packets shall have one and only one report block, corresponding to the data for the RIST sender.

The RR packet description presented below has been adapted from RFC 3550 sections 6.4.1 and 6.4.2:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|  RC=1   |   PT=RR=201   |            length=7           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     SSRC of packet sender                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Received stream SSRC                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| fraction lost |       cumulative number of packets lost       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           extended highest sequence number received           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      interarrival jitter                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         last SR (LSR)                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   delay since last SR (DLSR)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The fields are used as follows:

version (V): 2 bits
: Identifies the version of RTP, which is the same in RTCP packets as in RTP data packets. RIST packets shall have V=2.

padding (P): 1 bit
: Indicates whether or not there is padding at the end of the packet. RIST RR packets shall have P=0.

reception report count (RC): 5 bits
: The number of reception report blocks contained in this packet. RIST RR packets shall have RC=1.

packet type (PT): 8 bits
: Contains the constant 201 to identify this as an RTCP RR packet.

length: 16 bits
: The length of this RTCP packet in 32-bit words minus one, including the header and any padding. RIST RR packets shall have length=7.

SSRC of packet sender: 32 bits
: The synchronization source identifier for the originator of this RR packet.

fraction lost: 8 bits
cumulative number of packets lost: 24 bits
extended highest sequence number received: 32 bits
interarrival jitter: 32 bits
last SR timestamp (LSR): 32 bits
delay since last SR (DLSR): 32 bits

These fields shall be used as defined in RFC 3550 Section 6.4.1.

5.2.5 SDES RTCP Packets

RIST RTCP packets shall include one SDES packet. This SDES packet shall contain one item, the CNAME field. The complete SDES packet is shown below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|   SC=1  |  PT=SDES=202  |             length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     SSRC of packet sender                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    CNAME=1    |  name length  | user and domain name        ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| user and domain name (cont).  |0 to 3 bytes=0 |       0       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The fields are used as follows:

version (V): 2 bits

Identifies the version of RTP, which is the same in RTCP packets as in RTP data packets. RIST packets shall have V=2.

padding (P): 1 bit

Indicates whether or not there is padding at the end of the packet. RIST SDES packets shall have P=0.

source count (SC): 5 bits

The number of chunks contained in this packet. RIST SDES packets shall have SC=1, corresponding to a CNAME field.

packet type (PT): 8 bits

Contains the constant 202 to identify this as an RTCP SDES packet.

length: 16 bits

The length of this RTCP packet in 32-bit words minus one, including the header and any padding.

SSRC of packet sender: 32 bits

The synchronization source identifier for the originator of this SDES packet (the RIST sender or receiver).

CNAME identifier: 8 bits

Identifies this item as a CNAME. Always set to 1.

name length: 8 bits

Length, in bytes of the user and domain name field.

Note: The user and domain name is an ASCII string, which is **not** null-terminated as the length is specified in the previous field. RFC 3550 recommends that this string be programmatically generated in the form of "user@host". RIST implementations are free to use this field as they see fit. They may use the IP address of the sending or receiving device, in ASCII format (e.g., "192.168.129.10"). If the RIST device is multi-homed, it may use any of its local IP addresses.

A RIST device shall make no assumptions about the data in user and domain name field, but a vendor may make use of this for proprietary purposes. If a vendor does so, they shall have a means of disabling this extension.

The SDES packet shall be terminated with a set of bytes set to zero, as follows:

- There shall be at least one byte set to zero at the end of the SDES packet, not included in the CNAME name length.
- The overall size of the SDES packet shall be a multiple of 32 bits.

Note: Between 1 and 4 zero bytes will be added at the end of the SDES packet. These are not null terminators for the string - they indicate the end of the chunk list item in the SDES packet.

5.2.6 RTCP RTT Echo Request/Response Packets

The purpose of the RTCP RTT Echo Request/Response packets is to allow RIST endpoints to measure the Round Trip Time (RTT) to the remote endpoint. The RTT information can be used by receivers to optimize their retransmission requests. Support for this mechanism is optional. If this functionality is implemented, it shall operate as follows:

- The RTT Echo Request message includes a Timestamp. The receiving device shall echo the Timestamp in the RTT Echo Response without processing or interpretation. Upon receiving the RTT Echo Response, the originating RIST endpoint shall calculate the RTT by subtracting the Timestamp included in the message from the current time.
- The RTT Echo Response message includes a Processing Delay field. If the device receiving the RTT Echo Request cannot respond immediately, the device shall use this field to indicate how long it took to respond.
- The RTT Echo Request/Response messages may include optional padding, so that the RTT measurement can be made with packets of size similar to the stream packets.

The RTT Echo Request/Response packets shall be implemented with RTCP APP messages, as indicated below:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|0| Subtype |   PT=APP=204  |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  SSRC of media source                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         name (ASCII)                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Timestamp, most significant word               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Timestamp, least significant word               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Processing Delay (microseconds)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Padding bytes                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                              ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Padding bytes                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The fields are used as follows:

version (V): 2 bits

Identifies the version of RTP, which is the same in RTCP packets as in RTP data packets. RIST RTT Echo Request/Response packets shall have V=2.

padding (P): 1 bit

Indicates whether or not there is padding at the end of the packet. RIST RTT Echo Request/Response packets shall have P=0.

Subtype: 5 bits

This field identifies the type of the message. It shall be coded as follows:

RTT Echo Request: 2

RTT Echo Response: 3

Payload type (PT): 8 bits

This is the RTCP packet type that identifies the packet as being an Application-defined Message, with PT=204.

Length: 16 bits

The length of this packet in 32-bit words minus one, including the header and any padding. This is in line with the definition of the length field used in RTCP sender and receiver reports. If the packet has X padding bytes (where X is required to be a multiple of 4), then the value of this field shall be 5 + X/4.

SSRC of media source: 32 bits

The synchronization source identifier of the media source that this feedback request is related to. This field shall be used in the same manner as described in the Range-Based Retransmission Requests in section 5.3.2.2.

Name (ASCII): 32 bits

This field identifies the application. For RIST packets, it shall have the value 0x52495354, the ASCII codes for “RIST”.

Timestamp: 64 bits

The originator of this message (Subtype = 2) shall fill in an arbitrary value in this field, and the recipient of the message shall echo it back in the response (Subtype = 3). In order to aid debugging, the timestamp may be in NTP format: the Timestamp most significant word may be a value in seconds, and the Timestamp least significant word may be the fractional part. There is no requirement that this be the actual NTP time or that the nodes be NTP synchronized.

Processing Delay: 32 bits

In RTT Echo Request messages (Subtype = 2), the message sender shall fill this field with zeros, and the message receiver shall ignore the value. In RTT Echo Response messages (Subtype = 3), the message sender shall code its processing time in microseconds, expressed as a 32-bit unsigned integer in network byte order. The processing time is defined as the interval between the instant the RTT Echo Request message is received and the RTT Echo Response message is transmitted. The following considerations apply:

- Devices that respond as soon as possible may code zero in this field.
- Devices that purposefully delay a response shall use this field to indicate the amount of such delay.
- Devices are not required to have microsecond-precision in the delay measurement. They can use whatever clock precision they have, and express the final result in microseconds.

Padding: n*32 bits

The RTT Echo Request sender may want to measure the RTT for a packet of a certain size, so it may pad the packet with a number of additional bytes, with arbitrary content. The only constraints are that the number of padding bytes shall be a multiple of 4, and the resulting compound RTCP packet shall not exceed the link MTU. The RTT Echo Response packet shall have at least the same number of padding bytes as the corresponding RTT Echo Request packet, as long as the link MTU is not exceeded. The contents of the padding are arbitrary; the RTT Echo Response packet should echo the same padding bytes as received in the RTT Echo Request packet.

In a multicast environment, where a RIST sender is transmitting a stream to a number of RIST receivers, the meaningful RTT measurement is between each of the stream receivers and the sender. Multicast RTT Echo Requests are not useful unless there is a group of senders, and multicast RTT Echo Responses are never useful. In such a situation, if the stream is being transmitted to UDP port P, RFC 3550 requires that RTCP packets be transmitted to port P+1; therefore, all participants are listening to port P+1 for RTCP. Therefore, the following rules shall apply for multicast operation:

- The RTT Echo Request packet should be sent as a unicast to the participant for which the RTT measurement is desired (typically, the stream sender). If the stream is being transmitted to port P, the compound RTCP packet including the RTT Echo Request shall be sent to UDP port P+1. The participant's IP address is determined by inspecting the source IP address of the multicast packets received on port P.
- The RTT Echo Response packet shall be sent as a unicast to the source IP address of the RTT Echo Request packet, directed to UDP port P+1.
- Multicast group participants supporting RTT Echo Request/Response packets shall accept both multicast and unicast RTCP packets on port P+1.

5.3 NACK-Based Recovery Protocol

5.3.1 Protocol Overview (Informative)

RIST uses a NACK-based Selective Retransmission protocol to recover from packet loss. The general operation of this protocol is as follows:

- Once packet loss is detected, receivers request a retransmission of the lost packet or packets.
- Receivers implement a buffer to accommodate one or more network round-trip delays and packet re-ordering.
- A lost packet may be requested multiple times.

The receiver data flow is described below:

- Packets are received in a Reorder Section, which takes care of out-of-order packets. This mechanism also supports bonding of multiple channels (e.g., cell bonding).

- After the Reorder Section, packets cross into a Retransmission Reassembly Section. Packet losses are detected at the boundary of these two sections by looking at discontinuities in the RTP sequence number.
- The buffer works as a FIFO, and an arriving in-order packet will push packets one or more positions in the FIFO. Out of order packets with sequence numbers between the newest packet in the reorder buffer and the oldest packet in the retransmission reassembly buffer are placed in positions according to their sequence number in the buffer.
- The decision of where in the buffer packet loss is detected is left up to the implementation. An implementation targeted at providing minimum possible delay detects packet losses at the input of the buffer, and packets arriving out-of-order cause extra retransmissions. Conversely, an implementation supporting bonding of multiple links has a large enough reorder section to accommodate the worst case delay differential between the paths.

It is recommended that implementations make provisions for networks where short signal outages may happen. The buffer size will be a function of the round-trip time, packet jitter, and these outages, if present.

For the Simple Profile, the buffer size is manually configured at both sending and receiving ends.

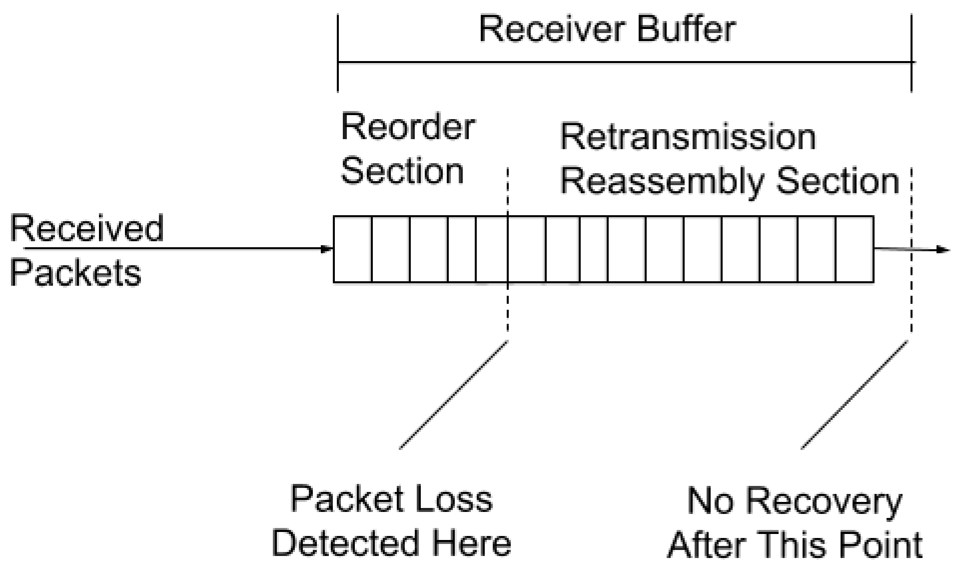

Figure 1: Receiver Buffers

5.3.2 Retransmission Requests

RIST Simple Profile includes two types of retransmission requests:

- A bitmask-based retransmission request, which is appropriate for individual packet losses and short loss bursts.
- A range-based retransmission request, which is appropriate for block losses.

RIST senders shall support both types of requests. RIST receivers may choose to implement either method, or both.

5.3.2.1 Bitmask-Based Retransmission Requests

Bitmask-based retransmissions shall be requested using the **Generic NACK** Message defined in RFC 4585 sections 6.2 and 6.2.1. The Generic NACK message specifies one or more ranges of 17 packets, using a bitmask to indicate which packets have been lost within each range. The message also contains the SSRC of the stream being requested. This is used to enable the sender to identify the flow in question. The retransmission request packet shall be formatted as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|   FMT=1  |      PT=205     |           length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  SSRC of packet sender                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  SSRC of media source                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:            Feedback Control Information (FCI)                 :
```

The fields are used as follows:

version (V): 2 bits

Identifies the version of RTP, which is the same in RTCP packets as in RTP data packets. RIST packets shall have V=2.

padding (P): 1 bit

Indicates whether or not there is padding at the end of the packet. RIST Generic NACK packets shall have P=0.

Feedback message type (FMT): 5 bits

This field identifies the type of the FB message and is interpreted relative to the type (transport layer, payload-specific, or application layer feedback). RIST messages shall use the Generic NACK code (1).

Payload type (PT): 8 bits

This is the RTCP packet type that identifies the packet as being an RTCP FB message. RIST messages shall use the code for Transport-Layer FB message (205).

Length: 16 bits

The length of this packet in 32-bit words minus one, including the header and any padding. This is in line with the definition of the length field used in RTCP sender and receiver reports. RIST messages shall use the value of **n+2**, where **n** is the number Generic NACK fields included in this packet, as specified below.

SSRC of packet sender: 32 bits

The synchronization source identifier for the originator of this packet. This field shall be ignored by the RIST sender.

SSRC of media source: 32 bits

The synchronization source identifier of the media source that this feedback request is related to. As indicated later in this document, the LSB of the SSRC is used to differentiate between original packets and retransmitted packets. The RIST receiver may use either value in the request packet.

Feedback Control Information (FCI): This field contains one or more instances of the 32-bit Generic NACK message shown below. Each FCI can request up to 17 lost packets. A Generic NACK message may contain multiple FCI fields.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            PID                |             BLP               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The fields in the FCI are:

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 following 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. Denoting the BLP's least significant bit as bit 1, and its most significant bit as bit 16, then bit i of the bit mask is set to 1 if the receiver has not received RTP packet number (PID+i) (modulo 2^16) and indicates this packet is lost; bit i is set to 0 otherwise. Note that the sender must not assume that a receiver has received a packet because its bit mask was set to 0. For example, the least significant bit of the BLP would be set to 1 if the packet corresponding to the PID and the following packet have been lost. However, the sender cannot infer that packets PID+2 through PID+16 have been received simply because bits 2 through 15 of the BLP are 0; all the sender knows is that the receiver has not reported them as lost at this time.

If a feedback message has multiple generic NACK fields, the fields should not overlap. In other words, any given sequence number, if included in the message, should fall into a single generic NACK field.

5.3.2.2 Range-Based Retransmission Requests

The range-based retransmission requests shall be implemented using RTCP. Ideally, this should be a Transport-Based Feedback Message (RTP type 205), with a new FMT value allocated by IANA. Such an allocation might be pursued by the RIST Activity Group in the future. In order to expedite the initial implementations of the protocol, the range-based retransmission request shall use an Application-Defined RTCP message (RTP type 204), shown below.

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |V=2|P| Subtype |   PT=APP=204  |             length            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                  SSRC of media source                         |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                         name (ASCII)                          |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |   Packet Range Requests                                       |
```

The fields are used as follows:

version (V): 2 bits

Identifies the version of RTP, which is the same in RTCP packets as in RTP data packets. RIST packets shall have V=2.

padding (P): 1 bit

Indicates whether or not there is padding at the end of the packet. RIST Generic NACK packets shall have P=0.

Subtype: 5 bits

This field identifies the type of the message. RIST range messages shall use code 0.

Payload type (PT): 8 bits

This is the RTCP packet type that identifies the packet as being an Application-defined Message, with PT=204.

Length: 16 bits

The length of this packet in 32-bit words minus one, including the header and any padding. This is in line with the definition of the length field used in RTCP sender and receiver reports. RIST messages shall use the value of **n+2**, where **n** is the number Range Request fields included in this packet, as specified below. A packet shall have a maximum of 16 range requests.

SSRC of media source: 32 bits

The synchronization source identifier of the media source that this feedback request is related to[1]. As indicated later in this document, the LSB of the SSRC is used to differentiate between original packets and retransmitted packets. The RIST receiver may use either value in the request packet.

Name (ASCII): 32 bits

This field identifies the application. For RIST packets, it shall have the value 0x52495354, the ASCII codes for "RIST".

Packet Range Requests: these are 32- bit fields, each requesting one packet range. The RTCP packet may contain multiple packet range requests. The packet range requests are shown below.

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Missing Pkt Sequence Start   | Number of addtl missing Pkts  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The fields in the Range Request are:

Missing Packet Sequence Start (16 bits):

RTP sequence number of the first packet dropped in the block

Number of Additional Missing Packets (16 bits):

Number consecutive packets being requested **after** the packet identified by the missing packet sequence start. For example, the Missing Packet Sequence Start is **N** and the Number of Additional Missing Packets is **A**, this indicates that packets from **N** to **N+A** inclusive have been lost. If **A** is zero, then only one packet (with sequence number **N**) is being requested.

5.3.2.3 RTCP Packet Size Considerations (Informative)

Both the Bitmask and Range RTCP NACK packets can contain multiple requests. It is a well-known fact that, when congestion occurs, smaller packets have a higher probability of being delivered. While this specification places no limits on how many requests can be included in a single RTCP NACK packet, it is recommended that implementers restrict the number of such requests. A limit of no more than 16 requests per packet is suggested.

5.3.3 Retransmitted Packets

For the retransmission of the requested data, the sender shall resend a copy of the requested packet, using the same transmission method as the other packets from this flow (i.e., the other packets with the same SSRC).

[1] RFC 3550 defines this field simply as SSRC/CSRC, and it was originally intended to contain the SSRC of the packet sender. However, this specification is re-defining this field as the SSRC of Media Source, since the SSRC of the packet sender is already available in the RR part of the compound RTCP packet, and is not useful in the RIST environment. Additionally, RFC 3550 indicates that APP packets with unknown names should be discarded, so non-RIST receivers can simply ignore this packet.

The sender shall use the SSRC field to identify the flow for which the retransmission is being requested. The retransmitted packet shall have the exact same sequence number and timestamp, and shall be identified by the least significant bit of the SSRC field, as follows:

- SSRC LSB=0: Original packet
- SSRC LSB=1: Retransmission packet

The remaining 31 bits of the SSRC shall be the same between the original and retransmitted packets. In a multi-flow situation, this allows the receiver to match retransmissions to original flows.

The retransmission packet shall be transmitted using the same method as the other packets in that same stream. If the flow is being sent to a specific IP address, then the retransmission packet shall also be sent to that address; if the flow is using bonding technology (being split and transmitted to multiple IP addresses in parallel), then the same algorithm that is used to determine the next IP address shall be used to determine where the retransmission packet will be sent. This ensures that, whatever the path, the receiver gets the copy of the packet. If the stream is being transmitted to a multicast address, all the other receivers may be able to benefit from receiving a copy.

5.3.4 Burst Control (Informative)

Packet bursts can cause additional packet losses in a typical network. It is recommended that RIST implementations manage packet bursts in the following two situations:

- NACK packet bursts (both Bitmask and Range modes): if a RIST receiver needs to send a large number of back-to-back NACK packets, care should be exercised in not creating too large of a burst. The most efficient mode (as measured by the number of NACK packets to be transmitted) depends on the loss pattern.
- Retransmitted packet bursts: a RIST sender can receive a NACK packet requesting a large number of retransmissions. It is recommended that implementations throttle the retransmitted packets so that the network is not overloaded.
  - Note: It is possible to generate a range-based retransmission request that requests the retransmission of every possible RTP sequence number (by setting the “Missing Pkt Sequence Start” field to any value, and the “Number of addtl missing Pkts” field to 65535).

For RIST Simple Profile, details of these techniques are left to the discretion of the implementer.

5.3.5 SSRC Filtering (Informative)

Both Bitmask and Range RIST packets contain the “SSRC of Media Source” field. Senders originating multiple streams can use this field to match requests with streams. A stream can usually be identified based on the IP addresses (source/destination) and UDP port of the RTCP packet, except when a source is sending multiple streams (differentiated by SSRC) using the same address and port. The stream identification strategy is left at the discretion of the implementer.

5.4 Bonding Support

RIST Simple Profile has optional support for bonding of multiple transmission channels (such as WiFi, LTE, etc.), in the following scenarios:

- An individual RTP media stream can be split between multiple channels in order to combine their bandwidths.
- An individual RTP media stream can be replicated between multiple network connections in order to increase reliability.

A network connection shall be defined as a distinct (destination IP address, destination UDP port) combination. The network topology is such that packets sent to all network connections arrive at the same RIST receiver.

Both techniques can be used simultaneously. RIST Receivers implementing bonding support combine the packets received from multiple channels in order to reconstruct the original media stream.

If bonding support is implemented, it shall operate as follows:

- A RIST receiver shall listen to multiple RTP and corresponding RTCP packets that make up a RIST stream over one or more network connections. The receiver shall re-aggregate the packets into one buffer, as described in section 5.3.
- The RIST receiver shall send RTCP packets to all network connections associated with a RIST stream, using the rules described in section 5.2.
- In case of packet loss, the RIST receiver may send the NACK RTCP packet on any of the connections associated with the RIST stream. The RIST receiver may also send the NACK RTCP over more than one network connection.
- If a RIST sender is replicating packets over multiple network connections, all copies of a given packet shall have the same RTP sequence number and timestamp.
- The RIST sender shall listen to NACK RTCP packets on all network connections.
- The RIST sender may choose any of the available network connections for packet retransmission. The RIST sender may also choose to send the packet retransmission over more than one network connection. The retransmitted packet shall be formed using the rules described in section 5.3.2.
- The RIST sender may use a combination of unicast and multicast destination addresses.

Note: A SMPTE-2022-7 Class-C compliant receiver will be able to receive a replicated RIST bonding stream (without the packet retransmission capabilities) if the Path Differential falls within the limits indicated for that class.

Appendix A - RIST Retransmission Request Examples (Informative)

The purpose of this section is to provide NACK packet examples to aid in preliminary implementations. In the examples below, numbers prefixed with 0x are hexadecimal, and numbers without that prefix are decimal.

For this example, assume that the RIST device is receiving a stream from IP address 192.168.1.10, UDP port 3000, SSRC 0xAABBCC00. The RIST device correctly receives packet sequence number 99, then misses packet 100, then receives packets 101 and 102, then misses the following 20 packets, from 103 to 122, and then receives all subsequent packets starting at sequence number 123.

The Bitmask-Based retransmission request packet will be sent to 192.168.1.10 port 3001 with the following contents:

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |V=2|0|   FMT=1   |      PT=205   |          length=4             |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |         SSRC of packet sender (ignored by RIST sender[2])        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |      0xAA     |       0xBB    |       0xCC    | 0x00 or 0x01  |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |          PID=100              |1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0|
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |          PID=117              |0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1|
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Range-Based retransmission request packet will be sent to 192.168.1.10 port 3001 with the following contents:

[2] RFC 3550 requires senders to keep an SSRC list, so an implementation that is fully compliant with that RFC will need to process this field. However, it is not required for RIST operation, so a RIST sender may choose to simply ignore this field.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|0|Subtype=0|   PT=APP=204  |           length=4            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      0xAA     |      0xBB     |      0xCC     | 0x00 or 0x01  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    0x52 (R)   |    0x49 (I)   |    0x53 (S)   |    0x54 (T)   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Start=100              |         Additional=0          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Start=103              |         Additional=19         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

In both examples above, the contents of the fields in **red** are fixed by this standard and never change.

Appendix B - Suggested Default Values (Informative)

RIST implementations complying with this specification are manually configured by the user. In the absence of user input, the following default parameters are suggested:

- Receiver Buffer: 1000 milliseconds
- Sender Buffer: equal or higher than receiver buffer
- Reorder Section: 70 milliseconds
- Number of Retransmission Requests per Packet: 7

The interval between retransmission requests can be derived from these parameters. It is the receiver buffer minus the reorder section divided by the number of retransmission requests. For the above values, the outcome is 132 milliseconds.