A Performance Measurement Study of the Reliable Internet Stream Transport Protocol

Ciro A. Noronha, Ph.D.
Cobalt Digital Inc.
Santa Clara, CA
ciro.noronha@cobaltdigital.com

Abstract - *The speed and reliability of the Internet has made it possible for broadcasters to use it as a cost-effective low-latency contribution link. Many companies have commercial products providing this functionality, and all of these products are implemented as a variation of the ARQ (Automatic Repeat reQuest) protocol. The Video Services Forum (VSF) has published the TR-06-1 Reliable Internet Stream Transport protocol specification to provide interoperability between such products. The first public demonstration of the protocol was performed during IBC 2018. After a short review of the RIST Specification and of the IBC 2018 demonstration, this paper presents a set of performance measurements for the protocol, using a commercially available RIST encoder/decoder pair, and network simulators to provide various types of signal impairment. The objective of the work is to provide configuration guidelines and performance bounds for broadcasters intending to implement the RIST protocol for contribution applications over the Internet.*

INTRODUCTION

Video transport over the Internet has been a reality for many years. Advances in compression technology have greatly reduced the bit rate required for good quality video, and infrastructure improvements have increased the available bandwidth and reliability of the Internet. The Internet is now a viable video contribution alternative to costly dedicated links.

However, packet delivery through the Internet is not guaranteed. Packets can be occasionally dropped, primarily due to instantaneous congestion in routers. Such packets need to be recovered for glitch-free video operation. Given enough time, any losses can be recovered, but contribution applications are typically latency-sensitive.

Experience has shown that the best technique to deal with such losses is one of the variants of the well-known Selective Retransmission method, called **A**utomatic **R**epeat re**Q**uest (ARQ). It represents a good tradeoff between latency and reliability.

There are a number of commercial products in the market that use ARQ to provide this functionality, using proprietary implementations that do not interoperate. In order to address this interoperability issue, the Video Services Forum (VSF) started the Reliable Internet Stream Transport (RIST) Activity Group in 2017 to create a common protocol specification, to promote interoperability between products from different vendors, and give broadcasters more choices when setting up an Internet link for contribution.

The first public RIST demonstration by the participating companies occurred in September 2018 during the IBC trade show, and the RIST Simple Profile Specification was published as VSF TR-06-1 in October 2018.

This paper starts with a review of VSF TR-06-1, RIST Simple Profile [1], and a description of the IBC 2018 demonstration. After that, performance measurement results of the RIST protocol using a network simulator are presented. Such results are then used to provide configuration recommendations for users.

RIST SIMPLE PROFILE

I. ARQ (Selective Retransmission)

RIST has selected the ARQ technique for packet recovery. This technique was devised in the 1960s and can be found in most computer networking textbooks (see [2] for an example). In general, the protocol works as follows:

- Sender transmits packets without waiting for any kind of feedback from the receiver.
- Packets have sequence numbers so that the receiver can identify packet losses.
- No acknowledgement is sent for packets that are correctly received.
- The receiver will request a retransmission for lost packets.
- A lost packet may be requested multiple times.

The process is illustrated in Figure 1, which shows an example of two successive losses. As soon as the receiver detects a packet loss, it will request a retransmission. At that point, it will need to wait for one network round-trip delay until that retransmission can possibly arrive. If it does not, then the packet may be requested again. If we denote the maximum number of retransmission requests by R and the network round-trip delay in seconds by T, it follows that both the receiver and the sender must have a buffer enough to hold RT seconds of content, and that the added latency of the protocol is RT. By controlling R, it is possible to control the latency-reliability tradeoff of the protocol.

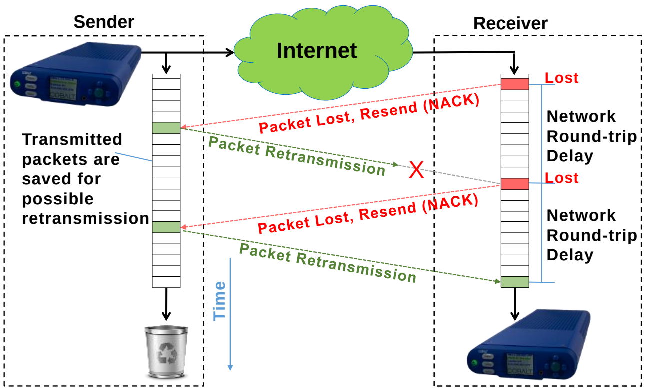

FIGURE 1: ARQ ILLUSTRATION

Figure 1 also indicates that the buffer at the sender side must hold at least *RT* seconds of content, to be able to satisfy the retransmission requests from the receiver. However, since the buffer at the sender side does not affect latency (packets are added to it after transmission), it can be made very large with no penalty other than memory consumption. Since the sequence numbers used for RIST are only 16 bits, it would take about 80 Mbytes of storage in the sender to cache the packets with every possible sequence number. This amount of memory is well within what is available in current systems, even in small embedded devices. Having a very large buffer at the sender simplifies overall system configuration.

II. RIST Simple Profile Protocol Description

RIST selected the Real Time Transport Protocol (RTP) [3] for the media transport. RTP is a very simple layer on top of UDP and provides sequence numbers (to detect packet loss) and timestamps (to remove network jitter, if required). The use of RTP ensures that RIST-compliant systems can interoperate with non-RIST systems at a base level without packet loss recovery.

RIST selected the Real Time Transport Control Protocol (RTCP) [3] associated with RTP for the retransmission requests. Two types of retransmission request messages have been defined:

1. **Bitmask-Based Retransmission Request:** Generic NACK message adopted from RFC 4585 [4]. This retransmission request covers a range of 17 consecutive packets, and can request any loss pattern within this range. It is useful for "salt-and-pepper" losses. One retransmission request can include multiple bitmasks. The format of this message is shown in Figure 2.
2. **Range-Based Retransmission Request:** This is implemented as an Application-Defined RTCP message ([3] section 6.7). It can request a continuous range of packets. One retransmission request can include multiple ranges. The format of this message is shown in Figure 3.

A RIST receiver may use either type of message. An advanced RIST receiver may dynamically decide which message to use based on the loss pattern, thus optimizing the bandwidth utilization. The "SSRC of Media Source" field helps the sender identify from which stream retransmission is being requested. This allows multiple streams to share the same UDP port at the sender.

32-bit

V=2	P	FMT=1	PT=205	Length
SSRC of Packet Sender				
SSRC of Media Source				
Missing Pkt Seq Number Start			Lost Packet Mask (1=loss)	
•••				
Missing Pkt Seq Number Start			Lost Packet Mask (1=loss)	

FIGURE 2: BITMASK-BASED NACK

32-bit

V=2	P	Subtype=0	PT=204	Length	
SSRC of Media Source					
0x52 ("R")			0x49 ("I")	0x53 ("S")	0x54 ("T")
Missing Pkt Seq Number Start				Number of additional lost pkts	
•••					
Missing Pkt Seq Number Start				Number of additional lost pkts	

FIGURE 3: RANGE-BASED NACK

The RTP specification [3] requires that senders and receivers periodically transmit RTCP packets. These packets are typically compound RTCP packets – i.e., multiple RTCP packets back-to-back in the same UDP payload. RIST uses this requirement to facilitate firewall configuration (described later in this section). The rules are:

- Sender transmits the media stream to UDP port P, where P is an even number and configured by the user.
- Sender transmits periodic RTCP messages to UDP port P+1, with source port S. Messages should be transmitted at least 10 times/second. RIST suggests the following message contents:
  - SR + SDES (CNAME)
  - Empty RR + SDES (CNAME)
- Sender listens for the RTCP messages on port S (as "response" to its RTCP messages). These will be compound RTCP messages that may or may not include the NACK messages described earlier. RIST only requires the sender to parse the NACK messages.
- Receiver listens for the media stream on port P and for the RTCP stream on port P+1.

- RIST does not require the receiver to do any processing of the content of the RTCP messages. The only data the receiver learns from these messages is their source UDP port (S – if not remapped by a firewall) and source IP address.
- Receiver must also send periodic RTCP messages, at least 10 times per second, addressed at the source UDP port and source IP address of the last received RTCP message from the sender. The minimum messages are:
  - RR + SDES (CNAME)
  - Empty RR + SDES (CNAME)
- If the receiver needs to request a retransmission, it sends a compound RTCP packet with one of the following formats:
  - RR + SDES (CNAME) + NACK
  - Empty RR + SDES (CNAME) + NACK

  Where NACK is one of the two NACK messages defined earlier.

In order to ensure protocol stability, it is necessary for the receiver to differentiate between original packets and retransmissions. RIST uses the SSRC field in the RTP header to make this differentiation, as recommended by RFC 4588 [5]. However, unlike RFC 4588, the retransmitted packet is an exact copy of the original RTP packet, except for the SSRC field. In order to simplify the association of an original packet flow with its retransmissions, RIST uses the following rules for SSRC:

- For original packets, the least significant bit of the SSRC is always set to 0 (zero).
- For retransmitted packets, the least significant bit of the SSRC is always set to 1 (one).

These choices allow maximum compatibility with non-RIST receivers. A receiver that filters by SSRC will simply ignore any retransmitted packets. A receiver that ignores the SSRC field may actually use the retransmitted packets based on their sequence numbers.

III. RIST and Firewalls

In the general case, the sender and receiver are behind firewalls for security reasons. RIST Simple Profile only requires that UDP ports P and P+1 be opened at the firewall located at the receive site. The sender is configured to transmit to the public IP address at the receiving site. The flow from the sender to UDP port P is unidirectional and will contain the audio/video content. The flow from the sender to UDP port P+1 establishes state in the firewalls at the sender and receiver sites; since the receiver directs its RTCP packets towards the source IP and UDP port of the traffic received in port P+1, these packets will be forwarded back to the sender. Since the RTCP flow from both sides is periodic, state is maintained in the firewalls. This is illustrated in Figure 4.

IV. Multicast Support

RIST Simple Profile includes IP Multicast support. Operation is very similar to unicast:

- Sender transmits media stream to UDP port P (even number) and a multicast IP address M.
- Sender transmits periodic RTCP packets to UDP port P+1, and the same multicast IP address M as the media stream.
- Receiver joins multicast M and listens on UDP port P for the media stream and port P+1 for RTCP.
- Receiver sends its RTCP packets to multicast M, UDP port P+1.
- Sender also joins multicast M and listens for receiver RTCP packets on port P+1.

Using this scheme, every receiver has the ability to “see” the RTCP packets from all other receivers, and can optimize its retransmission requests if desired (e.g., it may not request a retransmission that has been recently requested by another receiver).

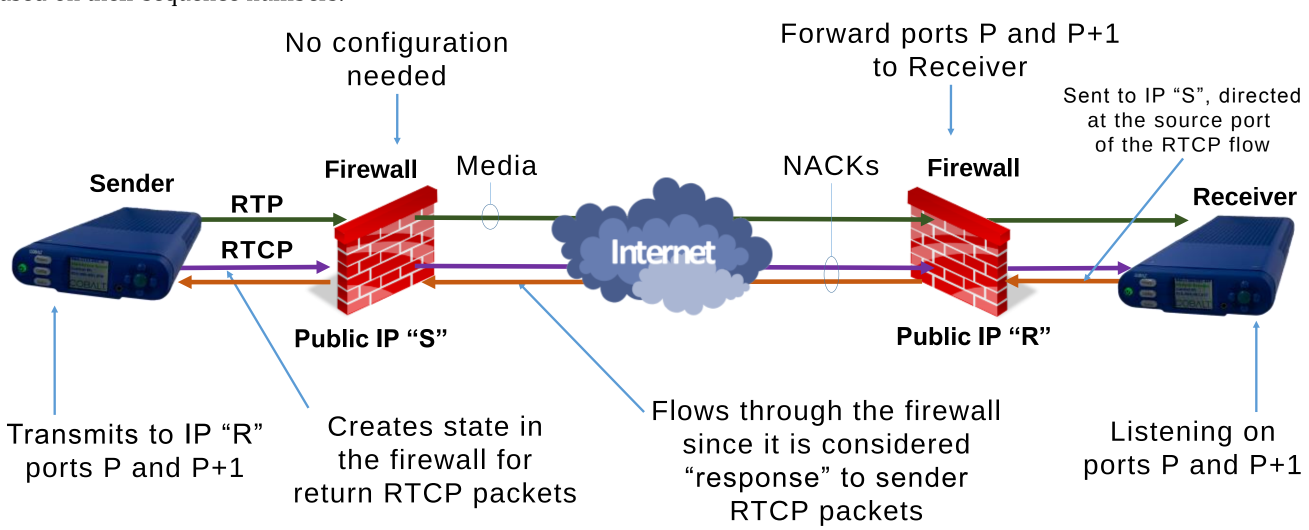

FIGURE 4: RIST AND FIREWALLS

V. *Bonding/Multi-Path Support*

RIST Simple Profile supports multiple paths between the sender and the receiver. This is applicable in the following scenarios (which are not mutually exclusive):

1. The media stream may be split over multiple lower-bandwidth paths (bonding). A typical case is the use of multiple cellular connections for media transmission.
2. The media stream may be replicated over two or more paths for reliability. This is similar to what is specified in SMPTE-2022-7 [6]. As a matter of fact, a SMPTE-2022-7 Class-C compliant receiver may be able to accept a multipath RIST stream depending on its buffer sizes.

In order to support re-ordering of packets, the RIST receiver needs to expand its buffer to include a re-ordering section. Packet loss is detected at the boundary of this buffer, as depicted in Figure 5 (reproduced from [1]).

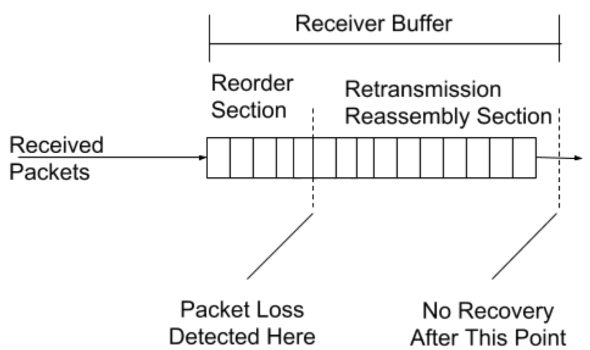

FIGURE 5: RIST RECEIVER BUFFERS

THE IBC 2018 DEMONSTRATION

A number of companies participating in the RIST Activity Group prepared a demonstration for IBC 2018. Each company implemented the protocol directly from the Specification (no shared libraries or code). A bank of RIST decoders were made available in Champaign, Illinois. The various participating companies transmitted streams over the Internet from different parts of the world to this bank of decoders. The signal from the decoders was combined in a multi-viewer, and the output of this multi-viewer was published live to YouTube. A short recording of the demonstration is still available for viewing [7]. Figure 6 shows a screen shot of the output of the multi-viewer.

As indicated in Figure 6, the streams were transmitted from Israel, USA-California, United Kingdom, USA-Virginia, USA-Florida, Canada, and USA-Massachusetts. This successful interop demonstration commenced two weeks prior IBC and proved that multi-vendor interoperable and reliable delivery over the internet can be achieved today.

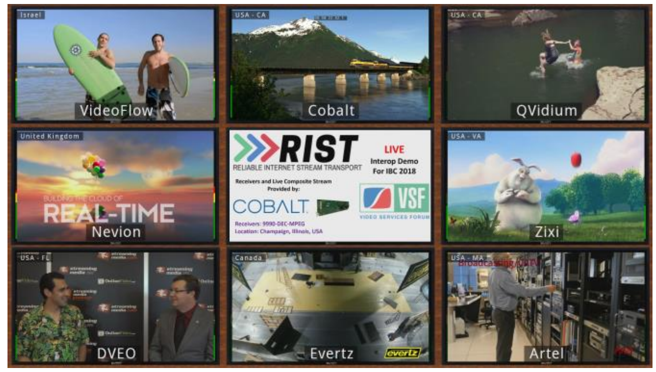

FIGURE 6: THE IBC 2018 DEMONSTRATION

RIST PERFORMANCE MEASUREMENTS

A set of measurements was performed using a real-time encoder transmitting actual audio/video content, a network simulator, and a real-time decoder. The use of a network simulator allows precise control of the network conditions. The setup is shown in Figure 7.

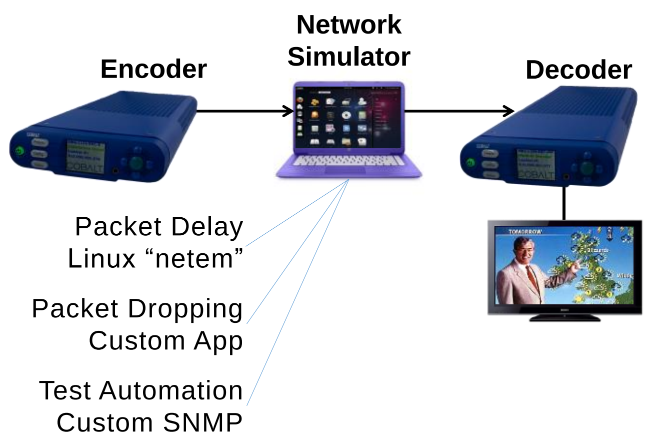

FIGURE 7: PERFORMANCE MEASUREMENT SETUP

The measurement parameters and procedure used were:

- Media bit rate: 8 Mb/s (1920×1080i59.54 source)
- Simulated round-trip delay: 200 milliseconds
- Random i.i.d. packet losses:
  - Single packet losses
  - 5-packet burst losses
- Two-minute runs
- Independent variable: number of retries, tested from 1 to 10
- Receiver retransmission buffer set to (200R + 100) milliseconds, where R is the number of retries
- Sender buffer set high enough to handle the worst-case receiver buffer
- For each retry value, increase the packet loss until at least one unrecovered packet is detected in the two-minute run.

- Record this packet loss rate
- Repeat each test 10 times

The results for single-packet losses are presented in Figure 8. The purple trace in the center represents the average across the 10 runs, and the blue and green lines above and below are the maximum and minimum values over the runs. In practical terms, if the network packet loss is known, the way to use Figure 8 is to read the number of retries required for that loss. For example, for 1% packet loss, the number of retries falls between 1 and 2. So, for this network performance, a minimum of 2 retries is needed. This implies that the Retransmission Reassembly Section of the receiver buffer must be at least twice the round-trip time.

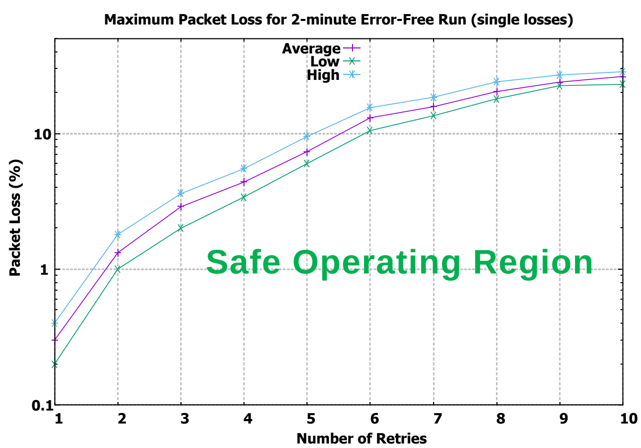

FIGURE 8: SINGLE PACKET LOSS MEASUREMENTS

The results for 5-packet burst losses are presented in Figure 9. The results are very similar to those presented in Figure 8 for single-packet losses, especially in regards to the average behavior. One artifact of this type of testing is that, to keep the same packet loss rate, burst losses have to be less frequent, thus making recovery somewhat easier at the lower packet loss values.

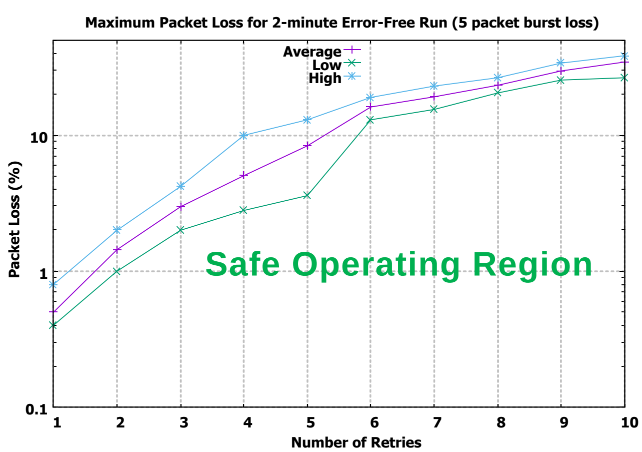

FIGURE 9: 5-PACKET BURST LOSS MEASUREMENTS

NOTES ON PACKET RE-ORDERING

In the previous section, the sizing and selection of the Retransmission Reassembly Section of the receiver buffer was discussed. The Reorder Section will now be considered. It is responsible for handling out-of-order packets in the network.

In practical terms, there is only one reason why packets may arrive at the destination out-of-order: when there are multiple network paths between the source and the destination. Routers may classify and transmit packets based on priority, but packets belonging to the same flow should receive the same classification.

The presence of multiple paths may be intentional (e.g., bonding – multiple end-to-end paths), or something that happens in the network backbone, outside of the control of the user. Service providers will likely route most packets through the same path; this path may change over time, and out-of-order instances will happen at the changeover.

The cost of adding a Reorder Section is added latency. Whether or not this matters depends on the application. Some applications, such as live news, require low latency. Other applications, such as monitoring, may not. The cost of not having a Reorder Section is bandwidth efficiency: the receiver may request the retransmission of a packet that is still in flight and will arrive shortly. Since packets have sequence numbers, receivers can identify and discard duplicates, so other than the waste of bandwidth, there is no other penalty.

In a bonding application, the user is aware of the multiple paths and has control over them. In this case, the round trip delays for each of the paths can be measured (using a simple utility such as “ping”) and the required size of the Reorder Section is simply the difference between the highest and the smallest round trip time, divided by two (since what matters is the one-way latency).

The more interesting question is what should be the size of the Reorder Section when the user has a single Internet connection for the sender and the receiver. One can always measure and characterize their individual link, but the great majority of users will not have the capability to do so, and most providers do not have data on out-of-order packets. Therefore, we turn to existing measurement data from [8]. In that paper, the authors characterize a number of backbone links, and measure, among other parameters, the amount of network-induced out-of-order packets in these links. In the paper, these events are called “Reorderings”. We summarize the relevant results in Table 1 below, with data from Table III in the original paper [8].

Table 1 indicates that, on average, for the links characterized by the authors, only 0.365% of the packets are out of order. Therefore, in the absence of any additional data, it may be reasonable to simply not have a Reorder Section. The penalty for that is a small increase in the retransmission data, but, as indicated in Table 1, this penalty is likely to be small.

TABLE 1: PACKET RE-ORDERING DATA (FROM [8])

	Total Packets	Reorderings	% Reorder
CDN	90,905,926	28,558	0.031%
Tier-1 ISP	39,403,671	307,615	0.781%
Tier-2 ISP	245,535,161	943,188	0.384%
OC48	153,143,822	653,717	0.427%
Total	528,988,580	1,933,078	0.365%

RIST USER RECOMMENDATIONS

When setting up a RIST Simple Profile link, the user will need to manually choose a few parameters to optimize the link. Our recommendations are:

- Find out the round-trip time between the sender and the receiver, using the "ping" utility.
  - If using bonding, do this for all links.
- If the network loss is known (e.g., there is an SLA in place), read the minimum number of retries from Figure 8. Common SLA values are:
  - 99% (1% loss): use 2 or more retries
  - 99.9% (0.1% loss): use 1 or more retries

  A safety margin is also recommended. For example, operation at 1% loss and 2 retries is marginal as it is at the border of the operating region in Figure 8. Add at least one retry for margin.
- If the network loss is not known, we recommend starting with 4 retries. Our experience is that 4 retries will give good results in most links. Indeed, the demonstration in Figure 6 was performed with 4 retries. Alternatively, if the application has a maximum latency requirement, divide that by the round trip time to find the number of retries, and use this value.
- If R is the number of retries selected and T is the round-trip time, the Retransmission Reassembly Section of the receiver buffer should be set to at least RT. If the application can tolerate it, we recommend a 10% additional margin as network delays tend to vary.
  - In a bonding situation, use the highest round-trip time for T.
- If the transmit buffer is configurable, it should be set as high as possible. At the very least, it must not be less than the receive buffer.
- If using bonding, the Reorder Section must be set to at least the difference between the highest and the lowest round-trip delay, divided by two. A safety margin is also recommended. If not using bonding, this can be left at zero.

When commissioning a link, it is always recommended that it be monitored for an initial period to validate the settings. The recommended adjustments are:

- If the receiver reports late packets, its buffers should be increased – the link latency is probably higher than expected.
- A certain number of duplicate packets is expected. However, it this number is significant, either increase the time between retries, or increase the size of the Reorder Section.
- If there are too many unrecovered packets, the number of retries should be increased if possible, with a corresponding increase in the Retransmission Reassembly Section of the receiver.

REFERENCES

[1] Video Services Forum TR-06-1, "Reliable Internet Stream Transport (RIST) Protocol Specification – Simple Profile", 2018-10-17

[2] Halsall, F., *Data Communications, Computer Networks and Open Systems*, Fourth Edition, 1996, pp 169-170

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

[4] Ott, J., Wenger, S., Sato, N., Burmeister, C., and Rey, J., "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback", *RFC 4585*, July 2006

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

[6] SMPTE 2022-7-2013, "Seamless Protection Switching of SMPTE ST 2022 IP Datagrams"

[7] Cobalt Digital (2018, September 28). *RIST Protocol Demo*. Retrieved from https://www.youtube.com/watch?v=dft8swbUjOk

[8] Jaiswal, S., Iannaccone, G., Diot, C., Kurose, J., and Towsley, D., "Measurement and Classification of Out-of-Sequence Packets in a Tier-1 IP Backbone", *IEEE INFOCOM 2003*, San Francisco, April 2003.