Preamble to Video Services Forum (VSF) Technical Recommendation TR-06-4 Part 7

August 27, 2026

The Reliable Internet Stream Transport (RIST) project was initiated as an Activity Group under the auspices of the Video Services Forum in 2017. The RIST Protocol is defined by TR-06-1 (RIST Simple Profile, published in 2018 and updated in 2020), TR-06-2 (RIST Main Profile, published in 2020 and updated in 2021 and 2022), and TR-06-3 (RIST Advanced Profile, published in 2021 and updated in 2022).

The TR-06-4 series of recommendations define ancillary features for the RIST protocol that are applicable to multiple profiles. This series includes:

- TR-06-4 Part 1, Source Adaptation, published in 2022.
- TR-06-4 Part 2, Use of Wireguard VPN in RIST Devices, published in 2023.
- TR-06-4 Part 3, RIST Relay, published in 2023.
- TR-06-4 Part 4, RIST Decoder Synchronization, published in 2024
- TR-06-4 Part 5, RIST Multicast Discovery, published in 2023.
- TR-06-4 Part 6, RIST Transport Stream Program Selection, published in 2024.

This document is TR-06-4 Part 7, RIST Satellite-Hybrid: In-Band Method. Satellite is the ideal way to distribute content to many geographically distinct locations. However, typical geo satellite distribution methods are unidirectional, and, depending on frequency used, may be subject to interference or rain fade. This Technical Recommendation describes a method to use the satellite (or any unidirectional transmission method based on MPEG-2 transport streams) as the primary distribution channel, while using RIST to recover any data that is corrupted or lost in transit. The method is backward-compatible with existing legacy receivers and requires the addition of synchronization data to the transport stream. It is called “in-band” due to this additional data.

TR-06-4 Part 7 was revised on September 5, 2025, to correct a minor error in Section 6.2, page 12.

TR-06-4 Part 7 was revised on December 10, 2025, as follows:

- Figures 1 and 2 have been modified to better reflect the operation of the TR.
- Section 5 has been updated with additional clarification on the system architecture.
- A new informative Appendix C has been added to discuss scalability options.

TR-06-4 Part 7 was revised on August 27, 2026, as follows:

- The marker definition was augmented to optionally include the locations of the NULL packets in the original transport stream prior to the modulator. This new feature allows the implementation of seamless protection switching as defined SMPTE ST 2022-7 between a satellite feed and one or more terrestrial feeds.

- The marker definition was augmented to optionally include additional descriptors to be managed by the Video Services Forum. This allows for backward compatible optional extensions to the marker.
- Appendix B was enhanced to include an informative PCR restamping method and further discuss an approach to implement seamless protection switching.

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. For additional information about the RIST Activity group, or to find out about participating in the development of future specifications, please visit Activity Groups – Video Services Forum.

Video Services Forum (VSF) Technical Recommendation TR-06-4 Part 7

Reliable Internet Stream Transport (RIST) Satellite-Hybrid: In-Band Method

Approved August 27, 2026

Executive Summary

Satellite distribution is the ideal way to send the same content to many locations that are geographically distributed. However, typical geo satellite distribution methods are unidirectional, and, depending on the frequency, may be subject to interference or rain fade.

This Technical Recommendation describes a method to use the satellite (or any similar unidirectional one-way transmission method employing MPEG-2 Transport Streams, such as DVB-T) as the main distribution channel, with RIST as a backup to recover data that is lost or corrupted in the space segment. The method is backward-compatible with existing legacy receivers and requires the addition of data to the transport stream sent to the satellite. It is called "in-band" due to this additional data.

Recipients of this document are invited to submit technical comments. The VSF also requests that recipients notify us 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.

Table of Contents

1 Introduction (Informative)

As broadcasters and others 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, incompatibility exists 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 using existing or new standards and recommendations.

Satellite distribution is the ideal way to send the same content to many locations that are geographically distributed. However, satellite transmission may be subject to localized degradation due to rain fade, interference, and other factors. Additionally, in many geo distribution cases, the satellite link is unidirectional, from one source to a multitude of receivers – no return channel exists. One possible approach to solving this content delivery problem is to augment satellite delivery using the Internet. The basic idea is to use the satellite for the "heavy lifting" (transmitting as much data as viable), with the Internet to "fill in the gaps". In other words, any data that is corrupted or lost in transit is retransmitted over the Internet using RIST, and only to the locations that need it. If there is a complete satellite fade (e.g., due to rain), the Internet can be temporarily used to deliver the complete signal. This way, if a region is experiencing any sort of fade or interference, only the receivers in that region need to use the Internet.

A requirement for the solution to this problem is that it must co-exist with current receivers to allow for gradual deployment. Such receivers expect a traditional MPEG-2 Transport Stream. The signal transmitted to the satellite either cannot change, or any changes to it must be backward compatible with existing legacy receivers. Broadcasters utilizing this solution can then gradually deploy the solution as needed, with high-priority sites being upgraded first.

This Technical Recommendation describes a method to use the satellite forward path in a way that is compatible with legacy receivers, and RIST to correct any lost data. It is called "in-band" because additional data is added to the satellite path. This solution is not limited to satellite; it can also be used to augment any large-scale point-to-multipoint network that uses a transport stream without an IP layer.

1.1 Contributors

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

Merrick Ackermans (CBS/Paramount)	Sergio Ammirata (SipRadius)	Paul Atwell (Media Transport Solutions)
Ruben Calzadilla (SES)	Eric Fankhauser (Evertz)	Eric Fankhauser (Evertz)
Ciro Noronha (Cobalt Digital)	Adi Rozenberg (AlvaLinks)	Wes Simpson (LearnIPVideo)

1.2 About the Video Services Forum

The Video Services Forum, Inc. (www.vsf.tv) 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 e-mail opsmgr@vsf.tv.

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

VSF TR-06-1:2020, Reliable Internet Stream Transport (RIST) Protocol Specification – Simple Profile

VSF TR-06-2:2024, Reliable Internet Stream Transport (RIST) Protocol Specification – Main Profile

VSF TR-06-3:2024, Reliable Internet Stream Transport (RIST) Protocol Specification – Advanced Profile

VSF TR-06-4 Part 2:2023, Reliable Internet Stream Transport (RIST) – Use of Wireguard VPN in RIST Devices

VSF TR-06-4 Part 3:2023, Reliable Internet Stream Transport (RIST) – RIST Relay

VSF TR-06-4 Part 5:2023, Reliable Internet Stream Transport (RIST) – RIST Multicast Discovery

VSF TR-06-4 Part 6:2024, Reliable Internet Stream Transport (RIST) – Transport Stream Program Selection

ISO/IEC 13818-1:2023, Generic coding of moving pictures and associated audio information, Part 1: Systems

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

SMPTE 2022-7:2019, Seamless Protection Switching of RTP Datagrams

Any mention of references throughout the rest of this document refers to the versions described here, unless explicitly stated otherwise.

4 Solution Requirements (Informative)

Satellites are the ideal solution to the problem of simultaneously distributing the same content to many geographically distributed receivers. Content is uplinked only once and retransmitted by satellite to all receivers at the same time. Adding a new receiver only requires putting up an antenna and pointing it; there is no additional burden or extra capacity required on the system.

However, satellite systems can suffer from degraded operation due to factors such as rain fade, interference, and similar issues.

For a broadcaster delivering signals to affiliates, occasional interference and outages are obviously not acceptable. Therefore, additional measures need to be taken to ensure reliable delivery.

The solution considered in this Specification is to augment the traditional satellite delivery with the Internet. Use the satellite for the "heavy lifting" bulk delivery, and "fix any remaining problems" with the Internet. If a site is suffering from interference or rain fade, send a recovery stream to that site only, and include in this recovery stream only the blocks of data that were lost or corrupted.

A diagram of the solution is shown in Figure 1. The solution is required to comply with the following requirements:

1. Satellite is the primary distribution method.
2. The satellite channel is unidirectional; there is no return channel.
3. The Internet is used only to recover dropped or corrupted data.
4. Data recovery is seamless with no glitches.
5. The solution needs to be capable of delivering the complete feed in case of satellite outage.
6. Optionally support ST 2022-7 seamless protection switching between a satellite path and one or more terrestrial paths.
7. The satellite signal is compatible with legacy receivers. More specifically, this means:
    a. The signal sent to the satellite is a traditional Transport Stream with one or more programs.
    b. Any information that is added to the Transport Stream is backward compatible with legacy receivers.

The following are specific requirements:

1. In many applications, the signal going to the modulator has been encrypted. Therefore, the solution cannot rely on specific fields in the transport stream payload.
2. Satellite modulators typically add or delete NULL packets for rate matching, and re-stamp PCRs to keep compliance. The solution cannot rely on the number and/or distribution of NULL packets, nor should it rely on the PCR values.

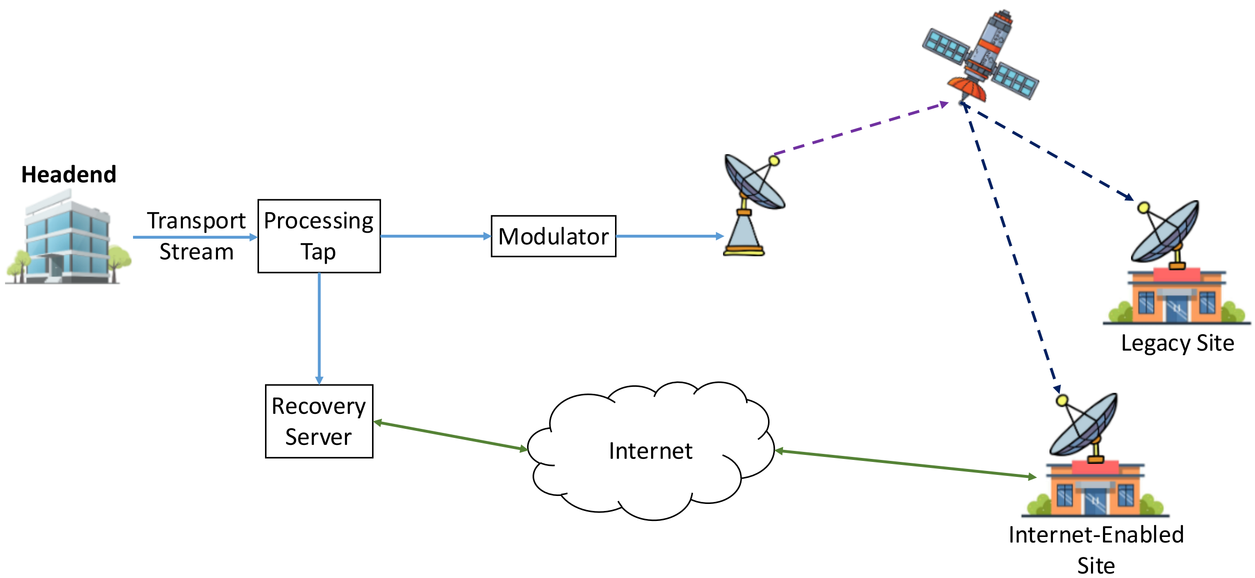

Figure 1: Solution Overview

5 Solution Architecture (Informative)

The basic solution architecture is shown in Figure 2. The operational flow is as follows:

1. The original Transport Stream goes through a Metadata Inserter, which adds metadata to the stream as described in Section 6. This metadata is multiplexed into the stream and correlates the transport packets in the original Transport Stream from the headend with corresponding RTP sequence numbers. The metadata allows an Internet-connected receiver to identify the RTP sequence number corresponding to lost or corrupted packets in the satellite stream.
2. The Recovery Server creates and buffers an RTP stream following the metadata information for sequence numbers and SSRC, as if it were sending the Transport Stream over RIST, using RIST Simple or Advanced Profiles.
   Note: the buffer at the recovery server needs to be large enough to accommodate the satellite round-trip latency, plus the worst-case Internet latency to all the receivers. This buffer size typically will be on the order of several seconds.
3. Internet-connected receivers synchronize the incoming satellite stream with the metadata. This synchronization yields the RTP sequence numbers required to request any lost or damaged data.

4. Lost packets are requested by Internet-connected receivers using the standard ARQ method in RIST Simple Profile or RIST Advanced Profile. Internet-enabled sites can keep a connection to the Recovery Server using RIST Main Profile or RIST Advanced Profile tunnels.
5. Optionally, Internet-connected receivers can use TR-06-4 Part 6 to select only a subset of programs they require, avoiding retransmission of unnecessary data.

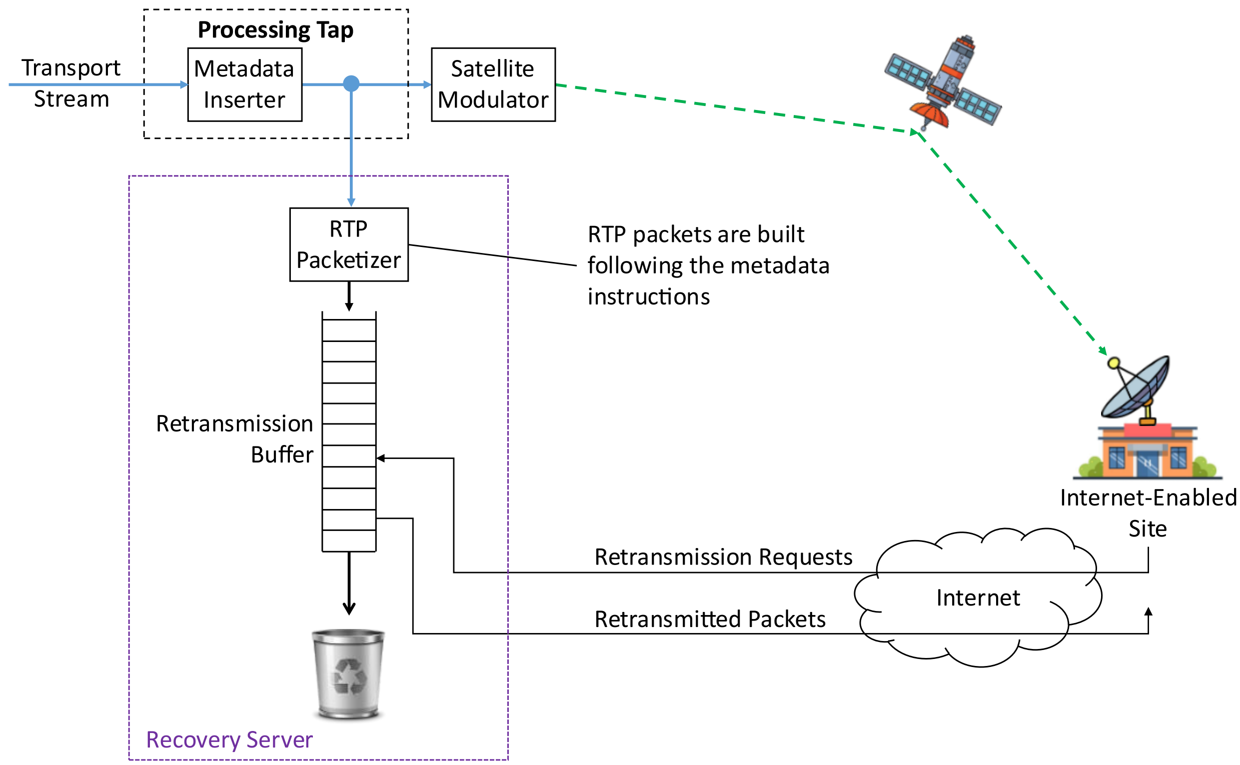

Figure 2: Solution Architecture

6 Metadata Creation Method

6.1 General Description

The in-band method described in this Specification relies on inserting metadata markers in the Transport Stream. The information contained in the markers allows the receiver to identify if any data is missing in the block delimited by the markers, and what RTP sequence number(s) it needs to request from the Recovery server to replace any missing data. If missing data is detected, the whole block between markers is requested.

Metadata markers shall be inserted at RTP payload boundaries; the number of payload boundaries between markers represents a tradeoff between overhead, retransmission traffic, and latency. As indicated in Section 5, RTP payloads shall consist of 7 transport packets, as specified in SMPTE 2022-2. The data in the marker shall contain the following information:

- Number of non-NULL packets since the last marker.
- Number of NULL packets since the last marker (to aid in PCR restamping).
- RTP sequence number of the first RTP payload in the block between markers.
- RTP sequence number for the first RTP payload in the next block.
- A marker sequence number.
- The SSRC of the RTP recovery stream in the server. The server shall choose an SSRC with the lowest significant bit set to zero as required by TR-06-1.

The data in the marker may contain the following information:

- A slot map with the locations of the NULL and non-NULL packets in the block.
- A set of optional extensions in Tag/Length/Value format.

The method is illustrated in Figure 3.

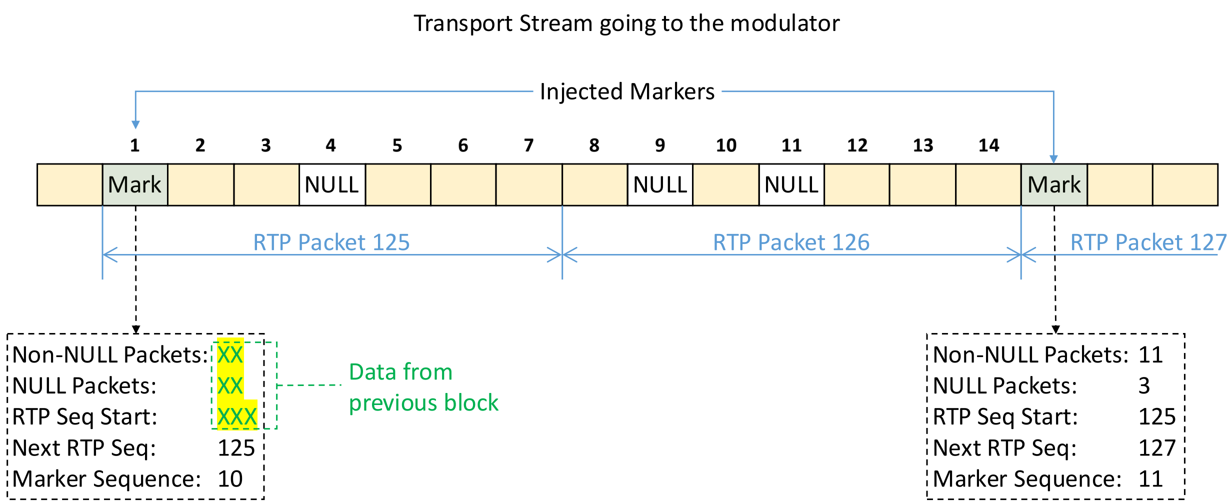

Figure 3: Metadata Insertion

In the example of Figure 3, markers are inserted every two RTP payloads. Upon reception of marker sequence 11, the Internet-enabled site shall compare the number of received non-NULL transport packets with the number reported in the marker (11). If the number is different, retransmission of the whole block (RTP packets 125 and 126) is requested.

6.2 Metadata Marker Format

The metadata marker shall use the **private_section()** syntax defined in ISO/IEC 13818-1, with **section_syntax_indicator** set to “0” (zero). The format for the private section is shown in Figure 4. The highlighted part is the private section header defined in ISO/IEC 13818-1.

Syntax	No. of Bits
private_section() {	
table_id	**8**
section_syntax_indicator	**1**
private_indicator	**1**
reserved	**2**
private_section_length	**12**
marker_sequence_number	**32**
non_null_count	**16**
null_count	**16**
rtp_sequence_start_msb	**16**
rtp_sequence_start_lsb	**16**
rtp_sequence_next_msb	**16**
rtp_sequence_next_lsb	**16**
source_ssrc	**32**
null_slot_map	**variable**
for (i=0; i < N; i++) {	
rist_extension()	
}	
CRC_32	**32**
}	

Figure 4: Metadata Marker Format

The fields in Figure 4 shall be defined as follows:

- **table_id (8 bits):** this field shall be set to 0xBF (user-defined, outside of the ISO, ETSI and ATSC ranges).
- **section_syntax_indicator (1 bit):** this field should be set to “0”. Receivers should be capable of parsing private sections with this bit set to “1” as defined in ISO/IEC 13818-1. Future versions of this Specification may include an option to fragment the marker section into multiple transport packets.
- **private_indicator (1 bit):** this field shall be set to “0”.
- **private_section_length (12 bits):** this field shall contain the number of remaining bytes in the private section, immediately following the field.
- **marker_sequence_number (32 bits):** this field shall be incremented by one at every metadata marker section transmitted. The Recovery Server may arbitrarily select the starting value.
- **non_null_count (16 bits):** this field shall contain the number of non-NULL packets in the previous block, including the metadata marker at the beginning of the block, and excluding the current metadata marker.
- **null_count (16 bits):** this field shall contain the number of NULL packets in the previous block, after insertion of the markers.

- **rtp_sequence_start_msb (16 bits):** if 32-bit sequence numbers are in use, this field shall be set to the 16-bit MSB of the sequence number for the first RTP packet in the previous block. If 32-bit sequence numbers are not in use, this field shall be set to zero.
- **rtp_sequence_start_lsb (16 bits):** This field shall be set to the 16-bit RTP sequence number of the first RTP packet in the previous block.
- **rtp_sequence_next_msb (16 bits):** if 32-bit sequence numbers are in use, this field shall be set to the 16-bit MSB of the sequence number for the first RTP packet in the block starting at this marker. If 32-bit sequence numbers are not in use, this field shall be set to zero.
- **rtp_sequence_next_lsb (16 bits):** this field shall be set to the 16-bit RTP sequence number of the first RTP packet in the block starting at this marker.
- **source_ssrc (32 bits): t**his field shall be set to the SSRC used in the RTP packets for this transport stream. If there are multiple uplinks, each with content that is different at the bit level, each uplink shall use a different SSRC value. Refer to Appendix C for discussion on how to use this field.
- **null_slot_map (variable):** this field may be inserted by the sender to indicate the location of the NULL and non-NULL packets in the block. Receivers may parse this field, if present, to exactly reconstruct the original sequence of NULL and non-NULL packets in the original transport stream. This field includes a bit for each transport packet in the block, set as follows:
    - "0": the transport packet contains non-NULL PID.
    - "1": the transport packet contains a NULL PID.

  The size of this field is:

  $$ceil\left(\frac{non\_null\_count + null\_count}{8}\right)$$

  Bits in this field are set sequentially starting from the most significant bit of the first byte in the field, which corresponds to the marker bit and thus shall be se to "0". Unused bits at the end of the field shall be set to "1" by the sender and ignored by the receiver.
  Example: in Figure 3, **non_null_count+null_count**=14, so the size of this field will be two bytes. Their values will be:

  ```
  byte 0:   00010000  (0x10)
  byte 1:   101000XX  (0xA3)
  ```

  The padding bits are indicated as "X", set to "1" by the sender and ignored by the receiver.
- **rist_extension():** if the sender inserts the **null_slot_map** field, it may insert zero or more optional **rist_extension()** fields. Receivers may parse these fields and process the supported ones.
- **CRC_32 (32 bits):** This field shall contain the CRC value that gives a zero output in the CRC decoder after processing the whole private section, using the algorithm specified in Appendix A of ISO/IEC 13818-1.

rist_extension() fields are defined as follows:

```
rist_extension() {
extension_type         8 bits
extension_length       8 bits
extension_value        8*extension_length bits
}
```

- **extension_type:** 8-bit code defining the extension. All extension values are reserved for assignment by the Video Services Forum and will be defined in a separate document.
- **extension_length:** size of the **extension_value** field, in bytes.
- **extension_value:** the contents of this field depend on the **extension_type** field.

Metadata marker private sections are encapsulated in transport packets. They are required to fit in a single transport packet. The fixed part of the marker takes 27 bytes out of the 183 bytes available for the section, so the total available size for **null_slot_map** and **rist_extension()** is 156 bytes.

The metadata marker PID should be set to PID 0x1FF0. Different PIDs may be used, and the value may be manually configured.

The metadata marker PID shall not be referenced in any of the PMTs in the transport stream.

6.3 Multiplexing the Metadata Marker in the Transport Stream (Informative)

Since the metadata marker needs to be inserted in very specific locations in the transport stream, the insertion cannot be achieved by simple NULL packet replacement. However, if the transport has enough available bandwidth in the form of NULL packets, the following simple algorithm can be used:

1. Define N as the "number of pending insertions so far". The starting value for N is zero.
2. The value of N is inspected as each transport packet is forwarded in the Processing Tap of Figure 2. The following actions are taken:
   a. If the transport packet is a NULL packet:
      i. If N is nonzero, the NULL packet is dropped and N is decremented by 1.
      ii. If N is zero, the NULL packet is forwarded.
   b. If the transport packet is not a NULL packet:
      i. If the transport packet does not have a PCR, it is forwarded.
      ii. If the transport packet has a PCR, the PCR value is re-stamped using equation (1) below, and the packet is forwarded.
3. Any time a metadata marker packet is inserted, N is incremented by one.

The PCR restamping equation is:

R: Transport stream bit rate, in bits/second.

N: Number of pending insertions defined above.

P_i: Incoming PCR.

P_R: Restamped PCR.

$$P_R = P_i + \frac{1504N}{R} \times 27{,}000{,}000 \tag{1}$$

In a transport stream with enough NULL packet bandwidth to accommodate for the metadata marker insertion, the value of N is expected to be small or zero most of the time. If N increments without bound, it is an indication that there is not enough NULL packet bandwidth to accommodate the markers. A calculation of the required marker bandwidth can be found in Appendix A.

7 RIST Message Extensions

One limitation with the method described in Section 6 is that a receiver that is turned on during a complete fade event does not have an initial RTP sequence number to request retransmissions using the RIST NACK messages. Additionally, if the satellite signal disappears completely, the receiver needs to have a mechanism to switch to the Internet feed. This section describes the following extensions of the RIST messages, created to address this situation:

- An extension for the receiver to ask the server to start transmitting a complete copy of the stream until otherwise turned off.
- An extension for the receiver to ask the server to stop transmitting a complete copy of the stream.

7.1 RIST Simple Profile Full Stream Request Messages

RIST Simple and Main Profiles use several Application-Defined RTCP messages, with different subtypes. The Full Stream Request Messages use two new subtypes, as shown in Table 1.

Table 1: RIST APP Subtypes

Subtype	Message	Specification
0	Range NACK Message	TR-06-1 Section 5.3.2.2
1	Sequence Number Extension Message	TR-06-2 Section 8.4
2	RTT Echo Request	TR-06-1 Section 5.2.6
3	RTT Echo Response	TR-06-1 Section 5.2.6
4	Reserved	
5	**Full Stream Request Enable**	**TR-06-4 Part 7 Section 7.1**
6	**Full Stream Request Disable**	**TR-06-4 Part 7 Section 7.1**

The format for the Full Stream Request messages is shown in Figure 5.

```
 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)                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: RIST Simple Profile Full Stream Request Message

The fields shall be set as follows:

version (V): 2 bits

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

padding (P): 1 bit

Indicates whether there is padding at the end of the packet. RIST Full Stream Request Packets shall have P=0.

Subtype: 5 bits

This field identifies the type of message. The following codes shall be used:
5: Enable Full Stream Transmission.
6: Disable Full Stream Transmission.

Payload type (PT): 8 bits

This is the RTCP packet type that identifies the packet as being an Application-defined Message. This field shall be set to 204.

Length: 16 bits

The length of this packet, expressed in 32-bit words minus one, including the header and any padding. This field shall be set to the value of **2**, since the size of Full Stream Request Message is three 32-bit words.

SSRC of media source: 32 bits

This field is normally set to the SSRC of the media source being requested. If the receiver does not know this value (e.g., at startup), it shall set to **0 (zero)**. Once the value is known, the receiver shall set this value to the actual SSRC of the media source.

Name (ASCII): 32 bits

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

RIST Simple Profile requires that the least significant bit of the SSRC in the RTP packets be set to zero for original packets, and one for retransmissions. RTP packets sent in response to the Full Stream Request Message defined in this section shall have the last bit of the SSRC set to zero. Any retransmissions of such packets (in case they are lost in transit over the Internet) shall be set to one.

A receiver requesting the full stream shall send a Full Stream Request message with Subtype=5 every 30 seconds as a keep-alive method.

Once the full stream is no longer needed by the receiver, it shall send a Full Stream Request Message with Subtype=6 every 5 seconds until the server stops transmitting the full stream.

The server shall stop sending the full stream if it receives a Full Stream Request message with Subtype=6, or if no Full Stream Request messages with Subytpe=5 are received for a period of two minutes.

If the server supports program selection (TR-06-4 Part 6), only the requested programs shall be transmitted in response to a Full Stream Request message.

7.2 RIST Advanced Profile Full Stream Request Messages

For TR-06-3 Advanced Profile, two new tunnel control messages (see Section 5.3) are defined. These two new tunnel control messages are indicated in Table 2.

Table 2: RIST Advanced Profile Control Index Values

Control Index	Message Type	Mandatory
0x0000	NACK Bitmask	
0x0001	NACK Range	
0x0002-0x0003	TR-06-4 Part 1 Link Quality Reports	
0x0004	TR-06-4 Part 4 Sender Synchronization Message	
0x0005	**Full Stream Request Enable**	
0x0006	**Full Stream Request Disable**	
0x0007-0x000F	Reserved for future NACK messages	
0x0010	RTT Echo Request	
0x0011	RTT Echo Response	Yes
0x0012-0x001F	Reserved for future RTT messages	
0x0020	ST 2022-5 FEC Row Packet	
0x0021	ST 2022-5 FEC Column Packet	
0x0022	ST 2022-1 FEC Row Packet	
0x0023	ST 2022-1 FEC Column Packet	
0x0024-0x002F	Reserved for future FEC messages	
0x0030-0x77FF	Reserved for future control messages	
0x7800-0x7FFF	Reserved for private vendor use	
0x8000	RIST Main Profile Keep-Alive message	Yes
0x8001	Flow Attribute message	
0x8002-0x800F	Reserved for future tunnel messages	
0x8010	Advanced Profile SRP authentication for PSK sessions	
0x8011	PSK Future Nonce Announcement Message	
0x8012-0x801F	Reserved for future authentication messages	
0x8020	Control Message Unsupported Response	
0x8021-0x802F	Reserved for future error messages	
0x8030-0x804F	TR-06-4 Part 3 RIST Relay Messages	
0x8050-0xF7FF	Reserved for future control messages	
0xF800-0xFFFF	Reserved for private vendor use	

The format for the two new control messages from Table 2 is shown in Figure 6.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Control Index = 0x0005/0006  |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     SSRC of Media Source                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: Full Stream Request Control Messages

The fields for the message in Figure 6 shall be set as follows:

Control Index: 16 bits

This field identifies the type of message. The following codes shall be used:
0x0005: Enable Full Stream Transmission.
0x0006: Disable Full Stream Transmission.

Length: 16 bits

Senders shall use this field to indicate the size of the Advanced Profile Control Message following the Length field, in bytes. This field shall be set to the value of **4**.

SSRC of media source: 32 bits

This field is normally set to the SSRC of the media source being requested. If the receiver does not know this value (e.g., at startup), it shall set to **0 (zero)**. Once the value is known, the receiver shall set this value to the actual SSRC of the media source.

RIST Advanced Profile requires that the R flag (see Section 52.3) be set to zero for original packets, and to one for retransmitted packets. Packets sent in response to the Full Stream Request Message defined in this section shall have the R flag set to zero. Any retransmissions of such packets (in case they are lost in transit over the Internet) shall have the R flag set to 1.

A receiver requesting the full stream shall send a Full Stream Request message with Control Index set to 0x0005 every 30 seconds as a keep-alive method.

Once the full stream is no longer needed by the receiver, it shall send a Full Stream Request Message with Control Index set to 0x0006 every 5 seconds until the server stops transmitting the full stream.

The server shall stop sending the full stream if it receives a Full Stream Request message with Control Index set to 0x0006, or if no Full Stream Request messages with Control Index set to 0x0005 are received for a period of two minutes.

If the server supports program selection (TR-06-4 Part 6), only the requested programs shall be transmitted in response to a Full Stream Request message.

Appendix A Number of RTP Packets Between Metadata Markers (Informative)

A.1 Bandwidth Overhead

The method described in this Specification adds one transport packet every K RTP packets, and each RTP payload carries 7 transport packets. The required data rate for metadata markers is:

R: Transport stream bit rate.
K: Number of RTP packets between the markers.
D: Required marker metadata bandwidth.

$$D = R\frac{1}{7K} \qquad (2)$$

If the marker packets are inserted using NULL packet replacement, the marker bandwidth needs to be lower than the NULL packet rate. Statistically multiplexed transport streams typically have on the order of 2.5% NULL packets, or an average of one NULL packet for each 40 transport packets. In such a case, K can be no less than 6, with a marker being inserted at every 6 × 7 = 42 packets, for an overhead of 1/42 = 2.38%.

For the Internet connections, it is expected that the receiving sites will use RIST Main Profile tunnels to the Recovery Server, and the baseline overhead is simply the Main Profile Keep-Alive messages. These require only around 1.5 kb/s regardless of the bit rate.

A.2 Latency and Retransmission Bandwidth

The receiver needs to buffer at least one full block between markers to determine if any data was lost. This adds a minimum processing latency corresponding to at least one block transmission time. However, since the marker packets themselves can be lost, this latency will be a multiple of the block transmission time. Marker loss is detected by the receiver using the marker sequence numbers. If a marker is lost, the receiver will need to request all the blocks between the received markers.

Note also that the granularity of the retransmissions is a block (unless the **null_slot_map** is included in the marker); if there is even one single error in the block, the whole block may need to be requested, depending on the error. Therefore, the size of the block represents a tradeoff between additional latency, satellite bandwidth overhead, and retransmission overhead. Increasing the size of the block between markers will decrease the satellite bandwidth requirement but will increase the receiver latency and the retransmission bandwidth. Implementers are encouraged to use the lowest block size supported by the target satellite feed.

If ST 2022-7 Seamless Protection Switching is being used, recovery can be done based on the terrestrial stream without any added latency.

Appendix B Receiver Configuration and Operation (Informative)

B.1 Receiver Buffer Sizing

Satellite links have relatively high latency, especially with geostationary satellites. This latency is typically higher than the round-trip latency through the Internet. For a receiver to be capable of seamlessly switching between an Internet stream and the satellite stream, while still being able to recover lost data, its buffer needs to be configured to be at least the satellite latency, plus a multiple of the Internet round-trip delay, plus a multiple of the block transmission time. This is especially important for sites that start up during a fade with an Internet stream; if they do not account for the satellite latency, they will not be able to switch to the satellite stream when it comes back, as it has more delay than the Internet version.

It is also technically possible to use this Specification to create a hybrid system with no added latency beyond the intrinsic satellite latency, by using the Full Stream Request functionality and accepting higher Internet bandwidth usage. This is based on the observation that satellite links suffer gradual degradation due to rain or sun activity. The receiver may monitor the satellite link quality and switch to Full Stream once it goes below a certain threshold. Since the propagation delay of the terrestrial link is likely to be less than the satellite path, the receiver can seamlessly switch to a terrestrial feed before it loses the satellite link. This method is less bandwidth-efficient than monitoring the markers for packet loss to request retransmissions but can provide recovery with zero additional delay added to the existing satellite delay. This obviously assumes that the terrestrial recovery network has been properly designed.

B.2 Receiver Connection Configuration

The configuration requirements depend on the RIST profile being used.

If the receivers are using RIST Simple Profile, then the configuration requirements are as follows:

1. The only configuration needed at the receiver site is the pair of UDP ports used to receive RTP and RTCP traffic.
2. If there is a firewall at the receiver site, it needs to be configured to expose the selected UDP ports.
3. The Recovery Server will need to be configured with a list of receive sites and their UDP ports.

An option to avoid having to configure the firewall is to use a Wireguard VPN as per TR-06-4 Part 2 and run RIST Simple Profile on the VPN. However, this does not change the requirement that the Recovery Server needs to be configured with a list of endpoints to send to. A possible alternative to configuring the server with a list of endpoints would be to select a multicast IP address and UDP port for the RTCP traffic and configure the server with that. Receivers send the RTCP packets to this multicast address and port; the server directs the return traffic to the source IP address of the RTCP packet, thus removing the need for a priori configuration.

If the receivers are using RIST Main or Advanced Profiles, the Recovery Server can be configured as a tunnel server, and the receivers all connect to it. This removes the need for firewall configuration at the receiver side and removes the need for the Recovery Server to be pre-configured with knowledge of all the receiver sites. The same mechanism suggested for Wireguard operation could also be used.

One option for scalability is to use the RIST Relay, described in TR-06-4 Part 3. In this case, the RIST Relay needs to be in Full Proxy mode (see TR-06-4 Part 3 Section 6.1) and needs to implement the NACK messages from Section 7.2. The receivers will connect to the RIST Relay instead of the Recover Server, and the Recover Server will simply send the full satellite feed to one or more RIST Relays.

If the Recovery Server is using a VPN that supports multicast, the multicast discovery method described in TR-06-4 Part 5 can be used as an alternative to the message extensions in Section 7.

B.3 Receiver Operation

Satellite modulators occasionally drop or insert NULL packets due to rate mismatch between their symbol rate clock and the incoming transport stream rate. These are nominally equivalent, but typically asynchronous to each other. The modulator will restamp the PCRs in the stream to compensate for this addition/deletion.

Since the marker insertion happens prior to the modulator, the receiver may encounter a block that has one extra NULL packet or is missing a NULL packet. If the receiver needs to correct this and recover the exact stream that went into the modulator, it will need to do the opposite operation – add a NULL if it is missing, or delete a NULL if there is extra, and re-stamp the PCR to match. This does not apply simply to that block; it applies to all future blocks.

A suggested high-level algorithm to restamp the PCRs at the receiver is:

1. For each PCR PID, record the PCR value, and start a TS packet counter. Let us call the recorded value P_s.
2. The next time a PCR is seen on this PID, do the following calculation:

$$P_E = P_S + \frac{1504N}{R} \times 27{,}000{,}000 \tag{3}$$

   here N is the number of TS packets since PCR P_s and R is the nominal transport rate.
3. Compare the P_E value calculated above with the received PCR, denoted by P_R. If the difference between P_E and P_R is more than ±500 ns, restamp the packet, replacing P_R with P_E.

4. Go back to step 1, recording the PCR value. Note that this applies to retransmissions as well.

If the receiver processes the **null_slot_map** and implements ST 2022-7 Seamless Protection Switching between a satellite stream and a terrestrial stream, it should be noted the method above is not good enough to ensure PCR jitter compliance between the original stream and the restamped stream. If that is desired, then the following modification can be made: the first time the receiver sees a new PCR PID, it requests a retransmission of that packet and uses the original PCR from the stream as the starting point of the algorithm above. This ensures that the recovered stream is compliant, and there is no offset from the original stream.

It should also be noted that the marker does not carry the original RTP timestamp. Since satellite streams are CBR, compliant RTP timestamps can be created as follows:

$$T_i = T_{i-1} + 90{,}000\,\frac{188\times 8\times 7}{R} \qquad (4)$$

Where T_i is the timestamp for packet i and R is the satellite channel bit rate.

It should be noted that, when performing ST 2022-7 seamless protection switching, the RTP timestamps for the satellite stream derived from equation (4) will not match the RTP timestamps of the corresponding packets from the recovery server. In other words, if one compares an RTP packet from the "blue" network with a packet with the same sequence number from the "red" network, these packets will have the same payload, but different timestamps. If the receiver is simply consuming the packets and disregarding the timestamp values, this difference can be ignored. However, if the receiver is further retransmitting the RTP packets, it needs to send consistent timestamps. In this case, the suggestion is to always use equation (4) and regenerate the timestamps.

Appendix C Redundancy and Scalability Considerations (Informative)

In a typical satellite deployment, there is a large number of receivers, each of which may require the full satellite signal in case of rain fade or any other outage. It is unlikely that a single Recovery Server will be able to service all the receivers. In such cases, it will be necessary to provide a bank of servers, all with access to the original stream, and these may even be geographically distributed and placed "closer" to a group of receivers.

The following options can be used in this case:

- All receivers are configured to contact the same server name or IP address, and a load balancer spreads the requests over multiple servers.
- Groups of receivers are configured to contact different servers. Receivers may be configured with a primary server and a fallback server.

If the deployment has redundant transmission chains and/or redundant uplinks, each transmission chain can be configured with a different **source_ssrc**. This way, receivers can identify which headend and transmission chain is online and can detect a change monitoring the value of the **source_ssrc** field and flush their buffers. Moreover, the receivers need to connect to servers that are handling the stream from the headend that is online. The following options can be used in this case:

- If there is a single network of servers, the headend that is online feeds this network of servers. The offline headend does not transmit out its stream.
- If each headend has a separate network of servers, the receivers are configured with the IP/name of each network of servers, and the corresponding **source_ssrc** value. A receiver that starts during a fade will contact a default server and switch to the correct one once the satellite signal is restored. Alternatively, the servers can be configured to only send periodic RTCP messages if they are online; this way the receivers can identify which server is currently online.