# Preamble to Video Services Forum (VSF) Technical Recommendation TR-06-4 Part 6

July 22, 2024

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.

This document is TR-06-4 Part 6, RIST Transport Stream Program Selection.  While RIST is content-agnostic, in some situations it may be used to carry an MPEG-2 Transport Stream.  Such a transport stream may include multiple PIDs and/or programs.  A RIST Receiver may only require a subset of such PIDs or programs, and bandwidth can be optimized if such information were made available to the sender, and if the sender only transmitted the required content.  This Specification provides an extension to the RIST Main Profile Keep-Alive message to convey such information.

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 http://vsf.tv/RIST.shtml.

# Video Services Forum (VSF)
# Technical Recommendation TR-06-4
# Part 6

## Reliable Internet Stream Transport (RIST)
## RIST Transport Stream Program Selection



Approved July 22, 2024

**INTELLECTUAL PROPERTY RIGHTS**

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

The Reliable Internet Stream Transport (RIST) is content-agnostic and, among other formats, supports the carriage of both Single-Program Transport Streams (SPTS) and Multi-Program Transport Streams (MPTS).  In the MPTS case, a RIST Receiver may require only a subset of the programs present in the MPTS.  If the RIST Sender can be made aware of which programs are required by the RIST Receiver, the sender can transmit only these programs to the receiver, thus saving network bandwidth.  This Specification defines a method by which this functionality can be implemented, using existing backward-compatible extensions to RIST Main Profile (VSF TR-06-2) and RIST Advanced Profile (VSF TR-06-3).

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, 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 using existing or new standards and recommendations.

RIST is content-agnostic, and it may be used to convey MPEG Transport Streams composed of either a single program (SPTS) or multiple programs (MPTS). A RIST Receiver may require only a subset of the programs in an MPTS, or a subset of the PIDs in an SPTS or MPTS. This Specification defines a mechanism by which the RIST Receiver can inform the RIST sender of which programs and/or PIDs it requires. The RIST Sender may optimize bandwidth by sending only the requested programs and using NULL Packet Deletion to keep compliant timing.

## 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/AMMUX) | Paul Atwell (Media Transport Solutions) |
|---|---|---|
| Ronald Fellman (QVidium) | Oded Gants (Zixi) | Brian Keane (Net Insight) |
| Holger Klaas (Nevion) | Ciro Noronha (Cobalt Digital) | Adi Rozenberg (AlvaLinks) |
| Wes Simpson (LearnIPVideo) | Thomas True (Nvidia) | |

## 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

**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

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

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

# 4    Program and PID Selection in RIST Main Profile

## 4.1   Introduction (Informative)

RIST Main Profile (VSF TR-06-2) includes support for a Keep-Alive message with an optional JSON payload.  Keep-Alive messages are sent periodically.  The message interval is between 1 second and 10 seconds.

This Specification defines an extension to the Keep-Alive JSON payload to provide program selection.  This Specification also defines the behaviors of a compatible RIST sender when processing the program selection.

A RIST Main Profile tunnel can carry multiple flows, and each flow can have multiple programs. The extension defined here supports per-flow program selection.

In a Transport Stream, programs are identified by the **program_number** field in the Program Association Table (PAT – see ISO/IEC 13818-1).  The PAT associates a **program_number** with a Program Map Table (PMT), which, in turn, lists all the PIDs included in the program.  The extension defined in this Specification allows the RIST Receiver to select programs using the **program_number**, which avoids the need for a-priori knowledge of individual PIDs.  Individual PID selection is also supported by this Specification.

## 4.2   Program Selection Extension to the Keep-Alive Message

RIST Main Profile only requires the transmission of a periodic Keep-Alive message, without any constraints as to its contents.  Devices compliant with this Specification shall support the Keep-Alive message described in TR-06-2 Section 5.6.3 and the JSON payload in Section 5.6.4. Moreover, devices compliant with this Specification shall set the **J** flag in the Capabilities Flags (TR-06-2 Section 5.6.3) to "1" on their transmitted Keep-Alive messages.

The JSON Keep-Alive message from TR-06-2 Section 5.6.4 is:

```
{
  "tunnelIP": "10.0.0.2",
  "remoteIP": "10.0.0.3",
  "excludedIP": ["192.168.1.0/24", "10.10.10.0/25"],
  "routing": true,
  "pskRotation": 600,
  "vendor": {
   "implementation": {
      "version": "2.3.5",
      "product": "Yellow RIST Machine",
      "vendorName": "RIST AG, Inc."
    },
    "features": null
  }
}
```

Devices compliant with this Specification shall support a new top level JSON key value, **contentSelection**. This key contains one or more program selection sections, which identify the flow to which the program selection applies, and the selection itself.

```
"contentSelection": [
  {
    "UDPPort": 5000,
    "DestinationIP": "239.1.1.1",
    "SourceIP": "192.168.1.1",
    "requestedPrograms": [1, 2, 3],
    "requestedPIDs": ["0x110-0x11F", "0x200"],
    "blockedPrograms": [4, 5, 6],
    "blockedPIDs": ["0x300-0x30F", "0x400", "0x500"]
  },
  {
    "UDPPort": 6000,
    "DestinationIP": "239.1.1.2",
    "SourceIP": "192.168.1.2",
    "requestedPrograms": [11, 12, 13],
    "requestedPIDs": ["0x210-0x21F", "0x300"],
    "blockedPrograms": [14, 15, 16],
    "blockedPIDs": ["0x400-0x40F", "0x500", "0x600"]
  }
]
```

The sender of the Keep-Alive message shall set the above parameters as follows:

- **UDPPort:** this field shall be set to the inner destination UDP port for the flow for which the program selection is being made.  The sender of the message shall always include this field.
- **DestinationIP:** if this field is present, it shall contain the inner destination IP address of the flow being selected.  If this field is omitted, the inner destination IP address shall not be used when determining to which flow this entry applies.  In Reduced Overhead mode, this field shall not be included by the sender of the message and shall be ignored by the receiver of the message.
- **SourceIP:** if this field is set by the sender of the message, it shall indicate the inner source IP address of the flow to which the message applies.  If this field is omitted, the inner source IP address shall not be used when determining to which flow this entry applies.  In Reduced Overhead mode, this field shall not be included by the sender of the message and shall be ignored by the receiver of the message.
- **requestedPrograms:** the sender of the message shall set this field to a list of one or more **program_number** values being requested.  If this field is not included, this shall indicate that the sender of the message is requesting all programs.
- **requestedPIDs:** if this field is included, it shall contain a list of PIDs being requested by the sender of the message.  The following formats are supported for the entries:
  - A single value in decimal, between `"0"` and `"8191"`.
  - A single value in hexadecimal, using the `0x` prefix, between `"0x0"`  and `"0x1FFF"`.
  - A range of decimal values, e.g., `"100-200"`.
  - A range of hexadecimal values, e.g.,  `"0x100-0x10F"`.
- **blockedPrograms:** if this field is included, it shall indicate which programs are explicitly not requested by the sender of the message.
- **blockedPIDs:** if this field is included, it shall contain a list of PIDs which are explicitly not requested by the sender of the message.  The format of this field is the same as the **requestedPIDs** field.

The receiver of the message shall process the program and PID selection as follows:

- If the message does not contain the **requestedPrograms**, **requestedPIDs**, **blockedPrograms**, and **blockedPIDs**, the receiver of the message shall assume that the sender is requesting all PIDs and all programs in the transport.
- If the message does not contain **requestedPrograms** but contains **blockedPrograms**, the receiver of the message shall assume that the sender is requesting all programs except the ones listed in **blockedPrograms**.
- If a given program is listed both in **blockedPrograms** and in **requestedPrograms**, the receiver of the message shall include the program in the transmission.

- If a given PID is listed both in **blockedPIDs** and in **requestedPIDs**, the receiver of the message shall include the PID in the transmission.
- If a given PID is to be included in the transmission because it is part of a program (listed in the program's PMT), but is also included in the **blockedPIDs** list, the receiver of the message shall not include the PID in the transmission.
- The receiver of the message shall ignore requests for inclusion of PIDs and programs that are not present in the selected transport stream.
- The receiver of the message shall ignore invalid items in the **requestedPrograms**, **requestedPIDs**, **blockedPrograms**, and **blockedPIDs** lists, including:
    o Programs outside the range of 1 to 65535.
    o PID values outside the range of 0 to 8191.
    o PID entries that do not correspond to valid decimal or hexadecimal values, as described earlier in this section.
    o PID ranges where the starting value is larger than the ending value.
- Keep-Alive messages are sent periodically.  If a Keep-Alive message does not include the **contentSelection** key, the receiver of the message shall assume that no changes in the current selection are requested.  If the value of the **contentSelection** key changes, the receiver of the message shall comply with the new value.

Once the Keep-Alive message sender makes a program selection, it should include that selection in every Keep-Alive packet until it detects that the selection has been implemented by the RIST sender.  At that point, it may remove the **contentSelection** key to save bandwidth.  The **contentSelection** key shall be transmitted at least once every 30 seconds.

In case of a tunnel disconnect and reconnect, the Keep-Alive message sender shall include the **contentSelection** key upon reconnection until the transport stream is re-established.

## 4.3  Program and PID Filtering Operation at the Sender
The sender shall comply with the following requirements:

1. If the original transport stream is Constant Bit Rate (CBR), its rate shall not change when different programs or PIDs are selected.  This may be accomplished by NULL Packet Deletion as per TR-06-2 Section 8.
2. Stream timing compliance shall be maintained.
3. The sender shall always include the following PIDs in the transport stream, if present. These take priority over the **blockedPIDs** list, i.e., they shall always be included even if blocked.
    - PID 0 (Program Map Table - PAT)
    - PID 1 (Conditional Access Table - CAT)
    - All the PMT (Program Map Table) PIDs listed in the PAT, including the non-selected programs.

- All the EMM (Entitlement Management Message) PIDs listed in the CAT.
4. If a program is included in the **requestedPrograms** list, the sender shall include all PIDs listed for that program in its PMT, including all the elements, PCR PID (if any), and any ECM (Entitlement Control Message) PIDs, except for the PIDs explicitly blocked by the **blockedPIDs** list.
5. The sender shall include all PIDs in the **requestedPIDs** list, if present.
6. The sender shall not include any PIDs listed in the **blockedPIDs** list, unless explicitly requested in the **requestedPIDs** list.

The sender may implement the above requirements by replacing all PIDs that are not selected for transmission by NULL packets, and then using the NULL Packet Deletion method from TR-06-2 Section 8 to reclaim their bandwidth.  However, the sender may use any other suitable method, provided that the requirements of this Section are satisfied.


# 5   Program and PID Selection in RIST Advanced Profile

RIST Advanced Profile (VSF TR-06-3) includes multiple payload types and can encapsulate RIST Main Profile packets.  The Program Selection function described in this Specification shall be implemented as described in Section 4, and encapsulated in RIST Advanced Profile in one of the following options:

- **Option 1:** The RIST Main Profile stream packets are encapsulated in RIST Advanced Profile using Type 8, as described in TR-06-3 Section 5.2.3.  This supports both Full Datagram and Reduced Overhead modes.
  OR
- **Option 2:** If using Reduced Overhead mode, the RIST Main Profile Reduced Overhead stream packets are encapsulated in RIST Advanced Profile using Type 3 as per TR-06-3 Section 5.2.3.
  OR
- **Option 3:** Using Direct Payload (Type 5) with a transport stream without any wrappers (Payload Format Descriptor set to 0x41AFD040), combined with payload compression as per TR-06-3 section 5.2.6.  In this case, the unselected PIDs shall be replaced by NULL packets, and the bandwidth gain is achieved by the payload compression.

In all cases, the Keep-Alive messages defined in this Specification shall be encapsulated as RIST Advanced Profile Control Messages (Type 4), with Control Index set to 0x8000, as described in TR-06-3 Section 5.3.6.